

1. 分布式实战（一）：

基于四台服务器实现lnmp分离部署，A服务器部署nginx，B服务器部署php，C/D服务器部署MySQL主从。

2. 实战环境：

```
A服务器 (nginx) : 192.168.75.133
B服务器 (php) : 192.168.75.134
C服务器 (master) : 192.168.75.135
D服务器 (slave) : 192.168.75.136
```

3. 部署流程：

3.1 部署nginx：

```
yum -C install nginx -y
```

3.2 部署网站代码：

```
tar xf wordpress-4.9.4-zh_CN.tar.gz -C
/usr/share/nginx/html/
chown nginx. -R /usr/share/nginx/html/
```

3.3 修改nginx配置文件：

```
location / {
    # 指定wordpress网站代码路径
    root    /usr/share/nginx/html/wordpress;
    index  index.php index.html index.htm;
}
# 省略部分信息
...
location ~ /\.php$ {
    #指定php-fpm远程服务器上wordpress代码路径
    root    /data/wordpress;
    # 指定php-fpm远程服务器ip
```

```
    fastcgi_pass    192.168.75.134:9000;
    fastcgi_index  index.php;
    fastcgi_param  SCRIPT_FILENAME
$document_root$fastcgi_script_name;
    include        fastcgi_params;
}
```

3.4 部署php:

```
yum install php php-fpm php-devel php-mysql -y
```

3.5 修改php-fpm配置文件:

```
[root@node4 ~]# egrep "(listen|user|group)" /etc/php-fpm.d/www.conf
listen = 192.168.75.134:9000
listen.allowed_clients = 192.168.75.133
user = nginx
group = nginx

# 启动php-fpm服务:
systemctl start php-fpm
# 查看进程/端口:
[root@node4 ~]# netstat -ntlp |grep 9000
tcp        0      0 192.168.75.134:9000  0.0.0.0:*
            LISTEN      1585/php-fpm: maste
[root@node4 ~]# ps -ef |grep php
root      1585      1  0 20:55 ?        00:00:00 php-fpm:
master process (/etc/php-fpm.conf)
nginx     1586    1585  0 20:55 ?        00:00:00 php-fpm:
pool www
nginx     1587    1585  0 20:55 ?        00:00:00 php-fpm:
pool www
nginx     1588    1585  0 20:55 ?        00:00:00 php-fpm:
pool www
nginx     1589    1585  0 20:55 ?        00:00:00 php-fpm:
pool www
nginx     1590    1585  0 20:55 ?        00:00:00 php-fpm:
pool www
root      1650    1037  0 21:00 pts/0    00:00:00 grep --
color=auto php
```

3.6 配置php远程服务器网站资源：

```
mkdir -p /data/  
rsync -av /usr/share/nginx/html/wordpress  
192.168.75.134:/data/
```

3.7 访问nginx：

要数据库连接信息，但是现在还没有配置数据库，所以需要先配置数据库。



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)

3.8 部署mysql：

```
# 主库和从库均执行以下命令：  
yum install mariadb mariadb-server mariadb-devel -y  
# 配置MySQL互为主从：  
[root@node5 ~]# egrep "log|server" /etc/my.cnf  
log-bin=node5-bin  
server-id=1  
  
[root@node6 ~]# egrep "log|server" /etc/my.cnf  
log-bin=node6-bin  
server-id=2  
  
# 启动主库和从库：  
systemctl start mariadb  
# 在node5上授权：
```

```
MariaDB [(none)]> grant replication slave on *.* to
"tongbu"@"192.168.75.136" identified by "123456";
MariaDB [(none)]> flush privileges;

# node6进行同步:
change master to master_host="192.168.75.135",
master_user="tongbu", master_password="123456",
master_log_file="node5-bin.000001", master_log_pos=477;
# 在node6上授权:
MariaDB [(none)]> grant replication slave on *.* to
"tongbu"@"192.168.75.135" identified by "123456";
MariaDB [(none)]> flush privileges;
#在node5进行同步:
change master to master_host="192.168.75.136",
master_user="tongbu", master_password="123456",
master_log_file="node6-bin.000001", master_log_pos=477;

# 在node5和node6上均开启slave进程, 然后执行show slave
status\G, 都能看到以下状态表示互为主从搭建成功!
slave start;
show slave status\G

Slave_IO_Running: Yes
Slave_SQL_Running: Yes

# 创建数据库:
MariaDB [(none)]> create database wordpress charset utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> grant all on wordpress.* to
"wordpress"@"192.168.75.134" identified by "123456";
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

3.9 配置数据库连接:

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="wordpress"/>	您的数据库用户名。
密码	<input type="text" value="123456"/>	您的数据库密码。
数据库主机	<input type="text" value="192.168.75.135"/>	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

3.10 wordpress部署成功:



4. 分布式实战（二）：

基于五台服务器实现Inmp分离部署，A服务器部署nginx，B/C服务器部署php，D/E服务器部署MySQL主从。

5. 实战环境：

A服务器 (nginx) : 192.168.75.133
B服务器 (php): 192.168.75.134
C服务器 (php) :192.168.75.136
D服务器 (master): 192.168.75.135
E服务器 (slave): 192.168.75.137

6. 部署流程:

6.1 配置nginx:

```
# 配置php均衡池:
upstream cgiwordpress {
    server 192.168.75.134:9000;
    server 192.168.75.136:9000;
}
location ~ /\.php$ {
    root          /data/wordpress;
    fastcgi_pass  cgiwordpress;
    fastcgi_next_upstream error http_500;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME
    $document_root$fastcgi_script_name;
    include      fastcgi_params;
}

# 记得重启nginx服务
```

6.2 配置php-fpm:

```
[root@node4 ~]# egrep "(listen|user|group)" /etc/php-fpm.d/www.conf
listen = 192.168.75.136:9000
listen.allowed_clients = 192.168.75.133
user = nginx
group = nginx

# 启动php-fpm服务:
systemctl start php-fpm
# 查看进程/端口:
```

6.3 同步wordpress代码:

```
rsync -av /data/wordpress 192.168.75.136:/data/
```

6.4 数据库授权:

```
grant all on wordpress.* to "wordpress"@"192.168.75.136"  
identified by "123456";  
flush privileges;
```

6.5 测试:


关闭其中一条php服务, 测试访问是否正常。

7. 报错解决:

An error occurred.

Sorry, the page you are looking for is currently unavailable.
Please try again later.

nginx连不上php远程服务器的9000端口, 检查远程服务器9000端口是否开启, 如果开启着, 则检查是否允许nginx连接。



File not found.

php资源没有找到。根据nginx配置文件, 查看php远程服务器上是否相应的网站代码。

nginx服务器检查php的location, 看路径:

```
location ~ /\.php$ {  
    # 远程服务器代码路径  
    root          /data/wordpress;  
    fastcgi_pass  192.168.75.134:9000;  
    fastcgi_index index.php;
```

```
        fastcgi_param  SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        include        fastcgi_params;
    }
```

php服务器端执行以下命令:

```
[root@node4 ~]# ls /data/wordpress/
index.php          wp-blog-header.php  wp-cron.php
wp-mail.php
license.txt        wp-comments-post.php  省略部分文件...
```