课程内容：

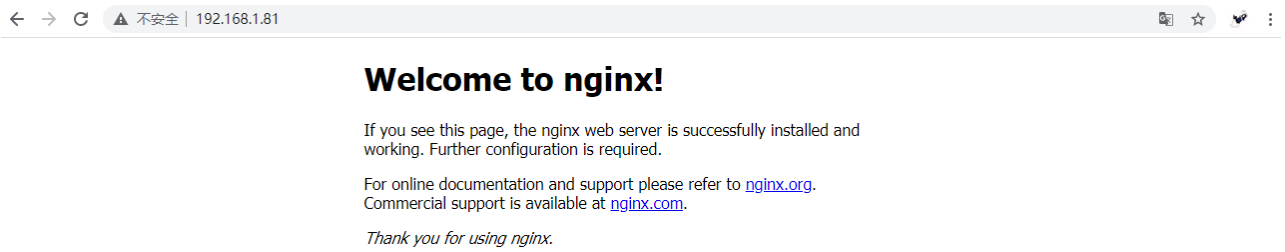
打造企业级的LNMP WEB架构实战

AWK、Sed、Grep分析nginx日志

SHELL变成脚本切割Nginx日期；

一、打造企业级的LNMP WEB架构实战：

当make源码编译方式构建nginx web平台，默认nginx web服务器只发布一套往回走哪，而且源代码是测试页面，访问如图所示



二、打造企业级LNMP web架构实战，

AWK sed grep分析nginx日志

shell变成教程切割nginx日志；

1、打造企业级LNMP web架构实战：

作为运维人员工作职责是：保障企业服务器，代码发布，网站更新，数据库，业务系统的维护等；

2、在企业中，开发人员开发了一套网站代码，基于PHP语言编写的，要求运维人员将PHP网站代码，实现外网用户的访问，作为运维人员该如何操作：

- 评估PHP网站后期的访问用户量（并发用户）
- 采购线上服务器（腾讯云主机，8C16G8M）8C核CPU，16G的内存，8M带宽
- 腾讯云主机操作系统（Centos7.x Linux操作系统）
- 注册外网域名，同时备案，cvc.net后期访问域名
- 构建LAMP、LNMMP WEB架构用于发布PHP网站；

```
//检查硬件环境：
[root@cdeba90ec46e /]# cat /etc/redhat-release
CentOS Linux release 7.6.2003 (Core)
[root@cdeba90ec46e /]# uname -a
Linux cdeba90ec46e 3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018 x86_64 x86_64
 x86_64 GNU/Linux
[root@cdeba90ec46e /]# df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay         100G  4.5G   96G   5% /
tmpfs           910M     0  910M   0% /dev
tmpfs           910M     0  910M   0% /sys/fs/cgroup
/dev/sda5       100G  4.5G   96G   5% /etc/hosts
```

```
shm              64M     0   64M   0% /dev/shm
tmpfs            64M  1.1M   63M   2% /run
tmpfs            64M     0   64M   0% /run/lock
tmpfs            64M     0   64M   0% /var/log/journal
tmpfs           910M     0  910M   0% /tmp
```

3、基于Centos7 Linux云主机，从0开始构建一套LNMP WEB架构，发布和处理PHP网站代码

LNMP：

L：基于linux内核研发的Linux操作系统（centos7）

N：Nginx开源，免费的高性能WEB服务器软件；

M：MySQL、Mariadb关系型数据库

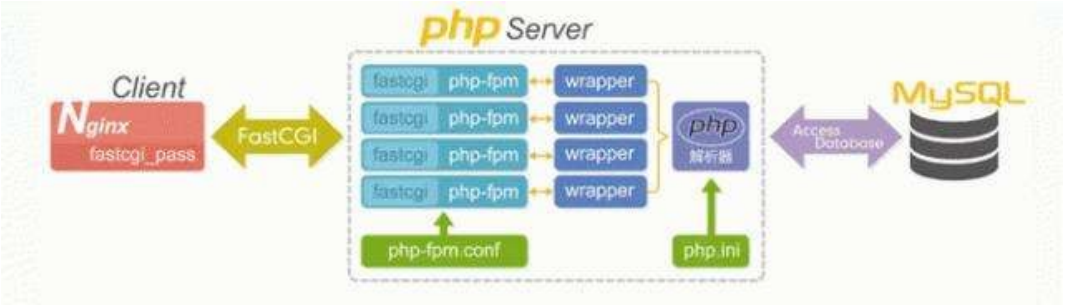
P：Perl、Python、PHP环境&PHP编译器

LNMP架构工作原理：

LNMP WEB架构中，Nginx为一款高性能WEB服务器，本身是不能处理PHP的，当我们收到客户端浏览器发送HTTP Regtest请求时，Nginx服务器响应并处理web请求，静态资源CSS，图片，视频、TXT等静态文件请求，nginx服务器可以直接处理并响应。

PHP动态页面请求nginx不能直接处理，nginx服务器会将PHP网页脚本通过接口传输协议（网关协议）传输给PHP-FPM（进程管理程序）

PPH-FPM调用PHP解析器（PHP-CGI）PHP解析器解析PHP脚本信息，最后PHP解释器将解析后的脚本放回到php-fpm，php-fpm在通过fase-CGI的形式将脚本信息传送给娘，如果所示；



注、docker环境下部署LNMP的基础环境：

```
[root@localhost ~]# docker run -tid --net=none --name=centos7-nginx.yum.82qiye
 --privileged=true centos7-ssh:zabbix-agent /sbin/init
27ba4cba71d5991e7cf814fb268c421474cd4018f0a62c560655d2bbb69c0634
[root@localhost ~]#  pipework  br0 centos7-nginx.yum.82qiye 192.168.1.82/24@192.168.1.1
[root@localhost ~]# docker exec -it centos7-nginx.yum.82qiye /bin/bash
```

三、从0开始构建LNMP WEB平台，主要有两种方式；

- YUM二进制方式
- MAKE源码编译方式

此处我们基于YUM二进制方式（网络源的形式，服务器能够上外网，配置局域网YUM源）构建LNMP的操作

1、部署nginx：

```
//1、添加Epel-release扩展源
[root@localhost ~]# yum -y install epel-release
[root@localhost ~]# ll /etc/yum.repos.d/|grep -aw epel
-rw-r--r-- 1 root root  951 Oct  3  2017 epel.repo
-rw-r--r-- 1 root root 1050 Oct  3  2017 epel-testing.repo
//2、安装nginx软件包：
[root@localhost ~]# yum -y install nginx    //显示Complete! 表示安装成功；
//3、检测nginx软件包是否安装成功
[root@27ba4cba71d5 ~]# rpm -qa |grep nginx
nginx-filesystem-1.14.1-9.module_el8.0.0+184+e34fea82.noarch
nginx-mod-http-perl-1.14.1-9.module_el8.0.0+184+e34fea82.x86_64
nginx-mod-mail-1.14.1-9.module_el8.0.0+184+e34fea82.x86_64
nginx-1.14.1-9.module_el8.0.0+184+e34fea82.x86_64
nginx-all-modules-1.14.1-9.module_el8.0.0+184+e34fea82.noarch
nginx-mod-http-xslt-filter-1.14.1-9.module_el8.0.0+184+e34fea82.x86_64
nginx-mod-stream-1.14.1-9.module_el8.0.0+184+e34fea82.x86_64
nginx-mod-http-image-filter-1.14.1-9.module_el8.0.0+184+e34fea82.x86_64
//4、查看端口和进程
[root@27ba4cba71d5 ~]# systemctl start nginx.service
[root@27ba4cba71d5 ~]# ps -ef |grep nginx
[root@27ba4cba71d5 ~]# netstat -tunlp |grep -aw 80
tcp        0      0 0.0.0.0:80  LISTEN      258/nginx: master p
tcp6       0      0 :::80       LISTEN      258/nginx: master p
```

2、部署Mariadb：

```
//1、安装Mariadb：
[root@27ba4cba71d5 ~]# yum -y install mariadb-server mariadb mariadb-devel
//2、检测Mariadb安装是否成功：
rpm -qa |grep mariadb
//3、查看Mariadb进程和端口号；
[root@27ba4cba71d5 ~]# systemctl start mariadb.service
[root@27ba4cba71d5 ~]# ps -ef |grep mysql
mysql        452       1  0 08:04 ?        00:00:00 /usr/libexec/mysqld --basedir=/usr
root         518     121  0 08:04 ?        00:00:00 grep --color=auto mysql
[root@27ba4cba71d5 ~]# netstat -tunlp |grep -aw 3306
tcp6       0      0 :::3306                 :::*                    LISTEN      452/mysqld
```

3、部署PHP：

```
//1、安装php
[root@27ba4cba71d5 ~]# yum -y install php php-devel php-fpm php-mysql
//2、检测PHP是否安装成功：
[root@27ba4cba71d5 ~]# rpm -qa |grep php
//3、查看PHP进程及端口号
[root@27ba4cba71d5 ~]# systemctl start php-fpm.service
[root@27ba4cba71d5 ~]# ps -ef |grep php
```

```
[root@27ba4cba71d5 ~]# netstat -tunlp|grep -aw 9000   //此时没有查到9000端口
```

4、根据如上LNMP部署指令操作，LNMP平台部署完成，查看其进程

```
[root@27ba4cba71d5 ~]# ps -ef |grep -wE "nginx|mysqld|php"
root         258       1  0 07:52 ?        00:00:00 nginx: master process /usr/sbin/nginx
nginx        259     258  0 07:52 ?        00:00:00 nginx: worker process
nginx        260     258  0 07:52 ?        00:00:00 nginx: worker process
mysql        452       1  0 08:04 ?        00:00:03 /usr/libexec/mysqld --basedir=/usr
root         598       1  0 08:14 ?        00:00:00 php-fpm: master process (/etc/php-fpm.conf)
apache       599     598  0 08:14 ?        00:00:00 php-fpm: pool www
apache       600     598  0 08:14 ?        00:00:00 php-fpm: pool www
apache       601     598  0 08:14 ?        00:00:00 php-fpm: pool www
apache       602     598  0 08:14 ?        00:00:00 php-fpm: pool www
apache       603     598  0 08:14 ?        00:00:00 php-fpm: pool www
root         647     121  0 08:23 ?        00:00:00 grep --color=auto -wE nginx|mysqld|php
```

9、要讲nginx和php-fpm进行配置整合，实现nginx检测到用户请求PHP动态网页时，nginx会将用户的请求通过CGI网关协议发送给后端PHP-FPM解释器取出来，nginx.conf配置代码如下：

```
//nginx配置：
[root@localhost nginx]#
        location / {                                        //第二步
            root   html;
            index index.php index.html index.htm;       //加上nginx.php表示引导页。

        location ~ \.php$ {
            root           /usr/share/nginx/html;   //更改发布目录  //第一步
            fastcgi_pass   127.0.0.1:9000;
            fastcgi_index  index.php;
            fastcgi_param  SCRIPT_FILENAME  $document_root$fastcgi_script_name;
                                                  //添加$document_root;表示发布目录  //第三步
            include        fastcgi_params;
        }
//查看nginx.conf文件并去掉#号空行。
[root@localhost nginx]# grep -vE "#|^$" nginx.conf
worker_processes  1;
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;              //以上为全局配置
    server {                            //以下为server主机的配置
        listen       80;
        server_name  localhost;
```

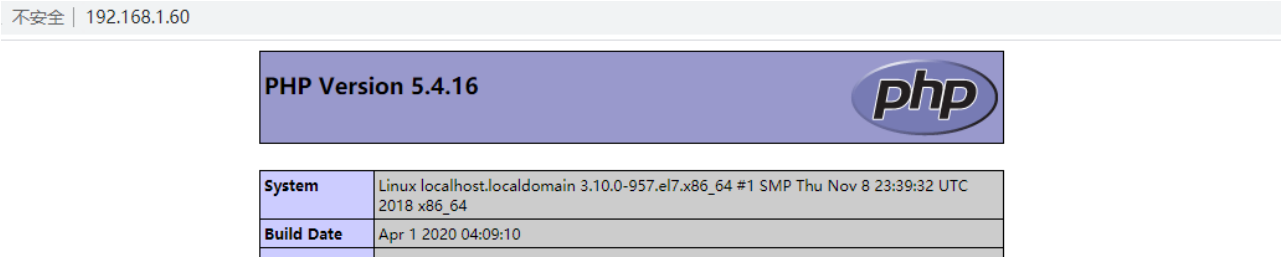
```
        location / {                    //location /是正常匹配，处于正则匹配后执行。
            root   html;
            index index.php index.html index.htm;
        }
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
        location ~ \.php$ {                         //location ~是正则匹配，是优先匹配，
            root           /usr/share/nginx/html;
            fastcgi_pass   127.0.0.1:9000;
            fastcgi_index  index.php;
            fastcgi_param  SCRIPT_FILENAME  $document_root$fastcgi_script_name;
                                //指定发布目录的变量，绝对路径也可以：/usr/share/nginx/html;
            include        fastcgi_params;
        }
    }
}
```

更改完成之后：

```
[root@27ba4cba71d5 nginx]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@27ba4cba71d5 nginx]# nginx -s reload
```

10、nginx、php-fpm发布目录；/usr/share/nginx/html,在该目录创建index.php，代码内容如下。

```
[root@27ba4cba71d5 nginx]# vim /usr/share/nginx/html/index.php
<?php
phpinfo();
?>
```



11、通过LNMP发布网站：

```
//我们只需要把我们打开发包解压到发布目录就可以了：
[root@localhost ~]# ls
anaconda-ks.cfg  web.html.tar
```

```
[root@localhost ~]# tar -zxvf web.html.tar /usr/share/nginx/html
```

三、AWK、Sed、Grep分析Nginx日志；

- 1作为运维人员，在企业中日志内容主要用于拍错，定问题，根据日志内容错误提示，能够第一时间去定位问题，拍错问题，从而快速的解决问题，降低企业的损失。
- 其实日志内容出了用于运维人员，开发人员，DBA排错之外，换可以对日志内容进行分析，统计，评估，从而掌握门户网站IP、PV、UV、访问量，资源分配，使用情况等。

```
[root@localhost nginx]# cd /var/log/nginx/    //cd到nginx的日志目录
[root@localhost nginx]# ll
total 1764
-rw-rw-r-- 1 nginx root 1527988 Nov  3 22:13 access.log
-rw-r--r-- 1 root  root   34543 Oct 31 11:45 access.log-20201031.gz
-rw-rw-r-- 1 nginx root   47309 Nov  1 10:12 access.log-20201101.gz
-rw-rw-r-- 1 nginx root   87592 Nov  2 13:24 access.log-20201102.gz
-rw-rw-r-- 1 nginx root   46903 Nov  3 12:41 access.log-20201103.gz
-rw-rw-r-- 1 nginx root       0 Nov  2 13:25 error.log
-rw-r--r-- 1 root  root    1488 Oct 30 13:20 error.log-20201031.gz
-rw-rw-r-- 1 nginx root     299 Nov  1 18:11 error.log-20201102.gz
[root@localhost nginx]# more access.log     //每一条都是一个用户的访问请求；
192.168.1.101 - - [03/Nov/2020:12:41:22 +0800] "POST /zabbix.php?action=notifications.get&sid
    =a2c13a51c1e3ec42&output=ajax HTTP/1.1" 200 436 "http://192.168.1.59/zabbix.php?action=da
shboard.view&ddreset=1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36"
```

1、基于SHELL编程三剑客Awk、Sed、Grep分析线上Nginx的日志，分析和统计Nginx全天总的请求数（访问量）操作的指令和方法如下：

```
//1、统计当天，总的用户访问量：
[root@localhost nginx]# awk '{print $0}' access.log |wc -l
                    //统计我们当天用户访问量；  单引号表示命令段，大括号把命令括起来，表示一个动作，
```

```
                    //print打印动作，$0表示文本所有内容，wc 统计，-l打印行号
3884
[root@localhost nginx]# sed = access.log|tail -2  // sed = 是显示行号的，tail -2表示显示最后两行
3884
192.168.1.101 - - [03/Nov/2020:22:22:45 +0800] "POST
/zabbix.php?action=notifications.get&sid=a2c13a51c1e3ec42&output=ajax HTTP/1.1" 200 436
"http://192.168.1.59/hosts.php?ddreset=1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36"

[root@localhost nginx]# sed = access.log|tail -2 |head -1      head -1 表示第一行
3884
[root@localhost nginx]# sed = access.log|tail -2 |head -1
3884
[root@localhost nginx]# wc -l access.log|cut -d "" -f1      //-f1便是fire字段
3884 access.log
[root@localhost nginx]# awk 'END{print NR}' access.log
3884
```

2、基于SHELL编程三剑客Awk、Sed、Grep分析线上Nginx的日志，分析和统计Nginx全天09:00~11:00之间总的请求数（访问量）操作的指令和方法如下：

```
[root@localhost nginx]# awk '/2020:09:00/' access.log|more
[root@localhost nginx]# sed -n '/2020:09:00/'p  access.log|more
[root@localhost nginx]# grep -aiw  '2020:12:41'  access.log|more
[root@localhost nginx]# sed -n '/2020:12:41/,/2020:12:00/'p  access.log|head -5
                                                    //范围 head表示前5条，tail表示末尾5条
[root@localhost nginx]# awk '/2020:12:41/,/2020:12:00/'  access.log|wc -l
3927
```

3、基于SHELL编程三剑客Awk、Sed、Grep分析线上Nginx的日志，分析和统计Nginx全天09:00~11:00之间总的请求数（访问量），将IP地址打印出来，同时将前20名的IP地址打印，将访问次数超过500的IP加入linux的黑名单；操作的指令和方法如下：

```
//都属于正则表达式：
//将09:00~11:00的IP地址打印出来；
[root@localhost nginx]#  sed -n '/2020:12:41/,/2020:12:00/'p  access.log|awk '{print$1}'|more
192.168.1.101
192.168.1.59
[root@localhost nginx]#
sed -n '/2020:12:41/,/2020:12:00/'p  access.log|grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}"|more
192.168.1.101
192.168.1.59
//将访问次数超过500次的IP加入linux的黑名单：
root@localhost nginx]# sed -n '/2020:12:41/,/2020:12:00/'p  access.log|grep -oE "([0-9]{1,3}\.)
>{3}[0-9]{1,3}"|sort -n|uniq -c|sort -nr|head -20              //打印访问量排前20名的IP地址；
   3964 192.168.1.101
   3961 192.168.1.59
注：sort -n正向排序；uniq -c去重并统计，sort -nr 逆向排序，head -20 打印20名；
```

```
16
17 //打印访问次数超过500次的用户地址；
18 [root@localhost nginx]# sed -n '/2020:12:41/,/2020:12:00/'p  access.log|grep -oE
19 >"([0-9]{1,3}\.){3}[0-9]{1,3}"|sort -nr|uniq -c|sort -nr|awk '{if(($1>500)) print$2}'
20 192.168.1.101
21 192.168.1.59
22 //访问量超过500次的加入IPtables防火墙黑名单：（IPtable）
23 [root@localhost nginx]# for ip in $(sed -n '/2020:12:41/,/2020:12:00/'p
24 >access.log|grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}"|sort -nr|uniq -c|sort -nr|awk '{if(($1>500))
25 >print$2}');do iptable -t filter -A INPUT -s $ip/32 -m tcp -p tcp --dport 80 -j DROP ;done
26                                                    //$ip/32表示IP地址，IPtable规则
```

Walter Savage Landor:strove with none,for none was worth my strife.Nature I loved and, next to Nature, Art:I warm'd both hands before the fire of life.It sinks, and I am ready to depart

——W.S.Landor
