### nginx反向代理软件

## 一、简介

nginx专为性能优化而开发的开源软件，是http及较好的反向代理软件，由俄罗斯的作者igor开发，优点是稳定和低资源消耗，以及对http并发连接的高处理能力，（3w-5w）



轻量级的web服务器软件。

Nginx在做反向代理时，提供性能稳定，并且能够提供配置灵活的转发功能。Nginx可以根据不同的正则匹配，采取不同的转发策略，比如图片文件结尾的走文件服务器，动态页面走web服务器，只要你正则写的没问题，又有相对应的服务器解决方案，你就可以随心所欲的玩。并且Nginx对返回结果进行错误页跳转，异常判断等。如果被分发的服务器存在异常，他可以将请求重新转发给另外一台服务器，然后自动去除异常服务器。

## 二、安装

1.安装依赖

yum -y install pcre-devel zlib-devel openssl-devel

创建用户，弃用nobody（权限最低的默认程序用户）

useradd -M -s /sibn/nologin nginx

5

tar -xf nginx-1.6.3.tar.gz -C /usr/src

3.配置

./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_flv\_module --with-http\_gzip\_static\_module --with-stream

--with-stream 支持四层转发

--with-http\_stub\_status\_module 启用模块以支持状态统计

--with-http\_ssl\_module 启用ssl模块

--with-http\_flv\_module 启用flv模块，提供寻求内存使用基于时间的偏移量文件

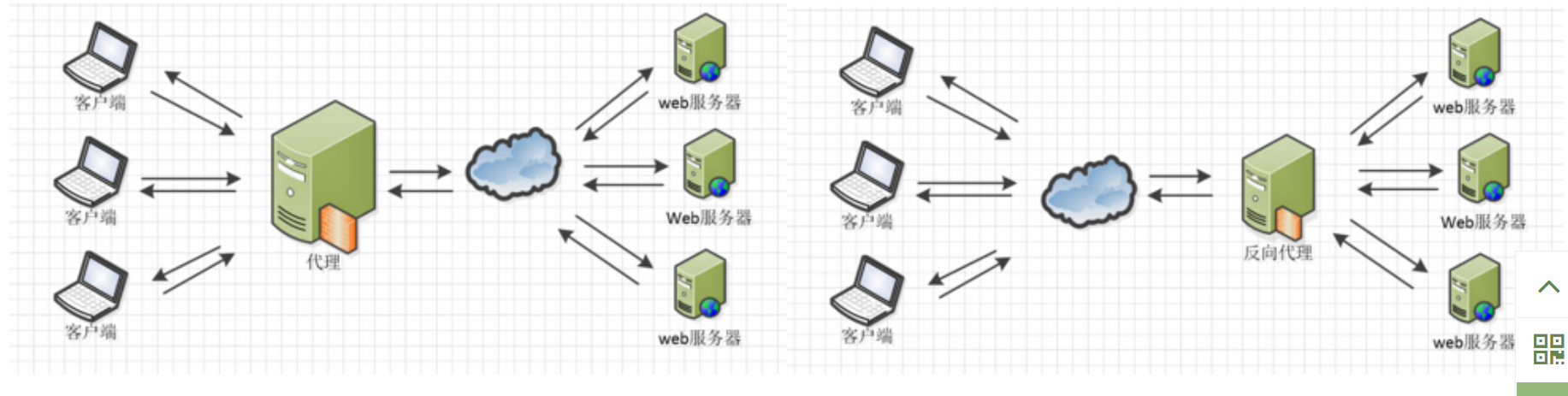
--with-http\_gzip\_static\_module 启用http传输压缩模块

4.make&&make install

5.创建软链接，为主程序nginx创建软链接

ln -s /usr/local/nginx/sbin/nginx /usr/local/bin/

## 三、正反向代理的区别



正向代理

作用：正向代理隐藏了用户，用户的请求被代理服务器接收代替，到了服务器，服务器并不知道用户是谁。

用途：为在防火墙内的局域网客户端提供访问Internet的途径

反向代理

作用：用户请求过多，服务器会有一个处理的极限。所以使用反向代理服务器接受请求，再用均衡负载将请求分布给多个真实的服务器。既能提高效率还有一定的安全性。  
用途：如果不采用代理，用户的IP、端口号直接暴露在Internet（尽管地址转换NAT），外部主机依然可以根据IP、端口号来开采主机安全漏洞，所以在企业网，一般都是采用代理服务器访问互联网。

## 四、nginx的工作原理

### 1.nginx采用异步非阻塞的工作模式

当I/O事件发生时，epoll就会告诉进程哪个事件由I/O事件产生，然后进程就会处理这个事件

### 2.处理过程

每进来一个请求，会有一个工作进程去处理，但不是全程进行处理，处理可能会发生阻塞的情况

比如：向后端转发请求，那个这个处理的工作进程不会一直等待，它会在发送完请求后，注册一个事件，等待后端服务器返回。

此时，再有新的请求，这个worker就很快按照这个方式处理；而一旦后端服务器返回信息，就会触发这个事件，worker就会进行处理，这个请求才会接着往下走；通过这种<快速处理并快速释放请求>的方式，达到同样的配置可以处理更大并发的目的。

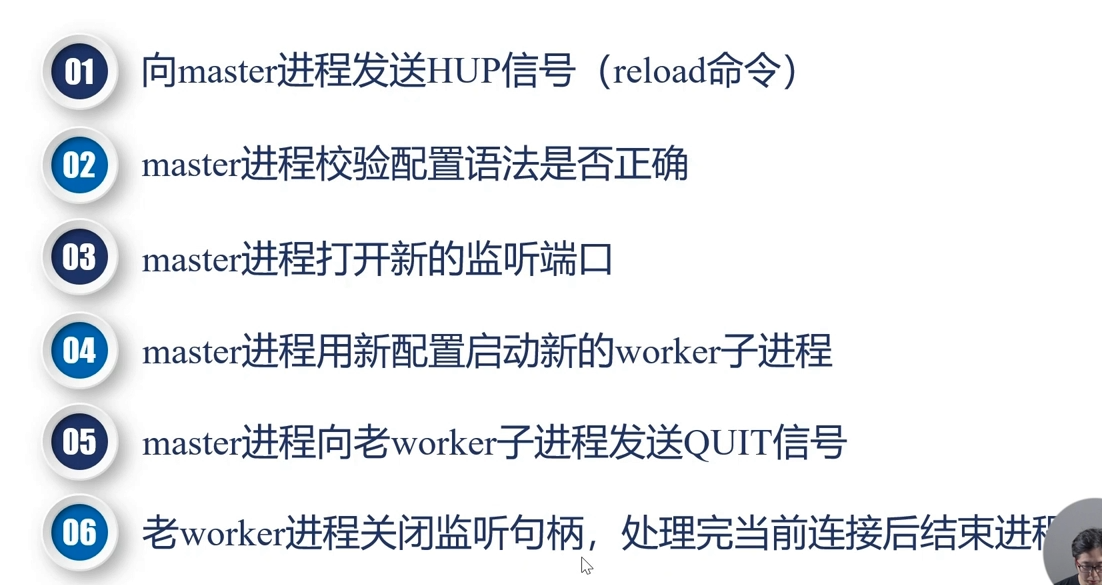
二、nginx的工作模式

1.master-worker模式

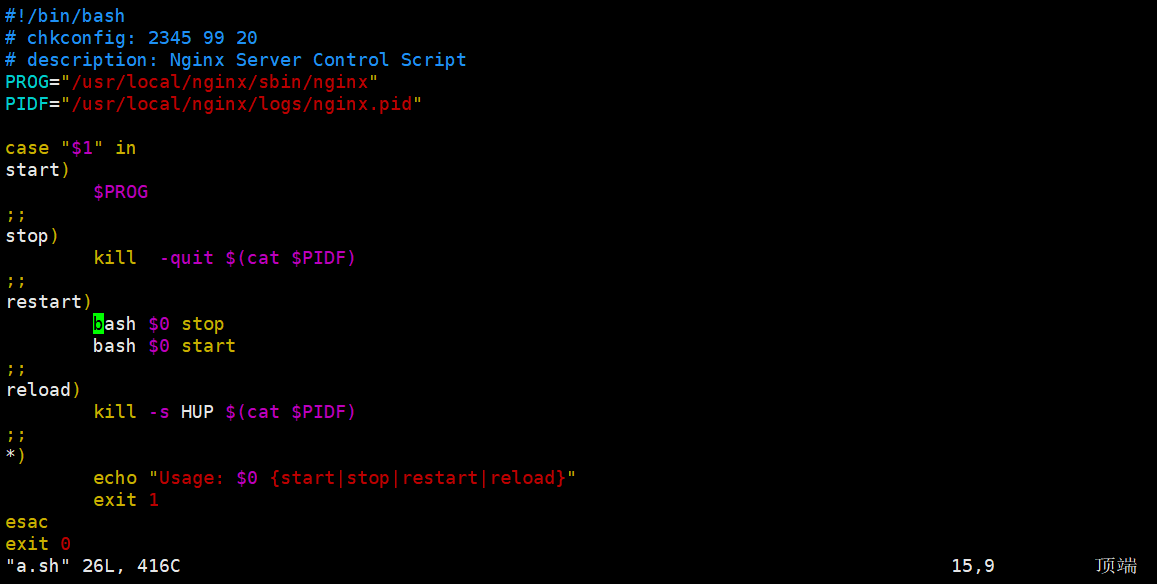
nginx启动成功后，会有一个master进程和至少一个worker进程；master进程负责处理系统信号，加载配置、管理worker进程；worker进程负责处理具体的业务逻辑；对于外部来说，真正提供服务的是worker进程；

优点：

稳定性高：一个worker进程挂掉后master进程会立即启动一个新的worker进程，保证worker进程数量不变，降低服务中断的概率；配合linux的cpu的亲核的匹配中，可以充分利用多核cpu的优势，提升性能；处理信号、配置重新加载等可以做到尽可能不中断服务



## 1、加入systemctl或者软链接快捷启动



#!/bin/bash

# chkconfig: 2345 99 20

# description: Nginx Server Control Script

PROG="/usr/local/nginx/sbin/nginx"

PIDF="/usr/local/nginx/logs/nginx.pid"

case "$1" in

start)

$PROG

;;

stop)

kill -quit $(cat $PIDF)

;;

restart)

bash $0 stop

bash $0 start

;;

reload)

kill -s HUP $(cat $PIDF)

;;

\*)

echo "Usage: $0 {start|stop|restart|reload}"

exit 1

esac

exit 0

chmod +x /etc/init.d/nginx

chkconfig --add nginx

重载配置

killall -s HUP nginx

nginx -s reload

退出进程 相当于 -3

kill -s QUIT nginx

## 2、配置文件结构

### 1、全局块：

配置影响nginx全局的指令。一般有运行nginx服务器的用户组，nginx进程pid存放路径，日志存放路径，配置文件引入，允许生成worker process数等。

### 2、events块：

配置影响nginx服务器或与用户的网络连接。有每个进程的最大连接数，选取哪种事件驱动模型处理连接请求，是否允许同时接受多个网路连接，开启多个网络连接序列化等。

### 3、http块：

可以嵌套多个server，配置代理，缓存，日志定义等绝大多数功能和第三方模块的配置。如文件引入，mime-type定义，日志自定义，是否使用sendfile传输文件，连接超时时间，单连接请求数等。

### 4、server块：

配置虚拟主机的相关参数，一个http中可以有多个server。

### 5、location块：

配置请求的路由，以及各种页面的处理情况。

### 位置：

/usr/local/nginx/conf/nginx.conf

### 详解：

### 1.全局块：

worker\_processes 1 进程数

worker\_cpu\_affinity 01 10; 亲核性（nginx性能调优）

双核四进程（01 10 01 10） 四核（0001 0010 0100 1000）

worker\_rlimit\_nofile 10240

同时连接的数量受限于系统上可用的文件描述符的数量，因为每个套接字将打开一个文件描述符。 如果NGINX尝试打开比可用文件描述符更多的套接字，会发现error.log中出现Too many opened files的信息。 ulimit -a 查看

ulimit -n 10240(临时设置)

error\_log logs/error.log 全局错误日志文件位置

pid logs/nginx.pid pid文件的位置

### 2.I/O envents块

使用“ecvents{}”界定标记，用来指定nginx进程的I/O响应模型，每个进程的连接数等设置

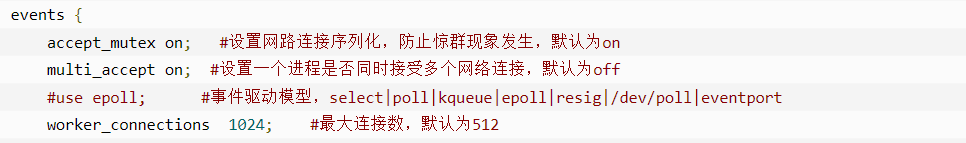
events{

use epoll //对2.6以上的内核，建议使用epoll模型以提高性能

性能

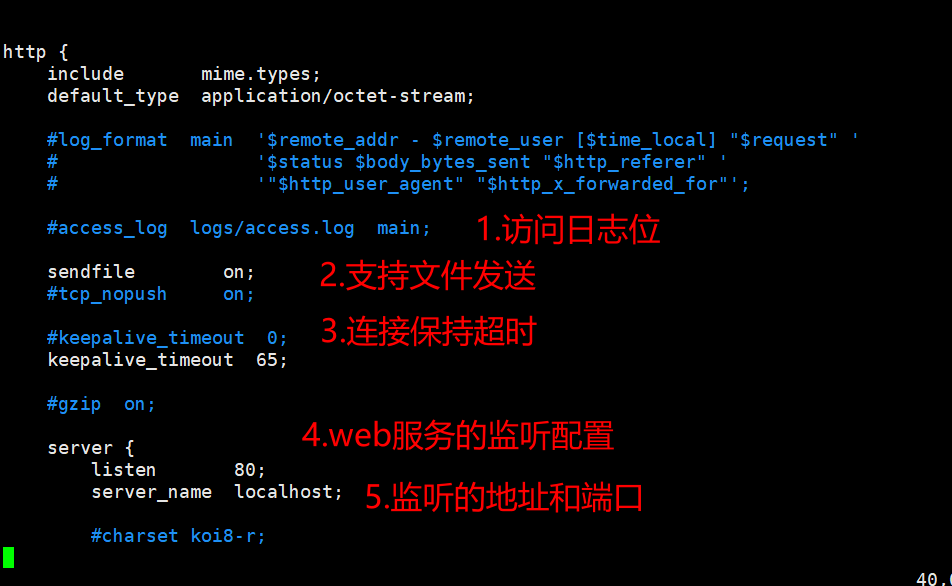
worker\_connection 4096 每个进程允许的最多连接数（默认为1024），最高可能10000

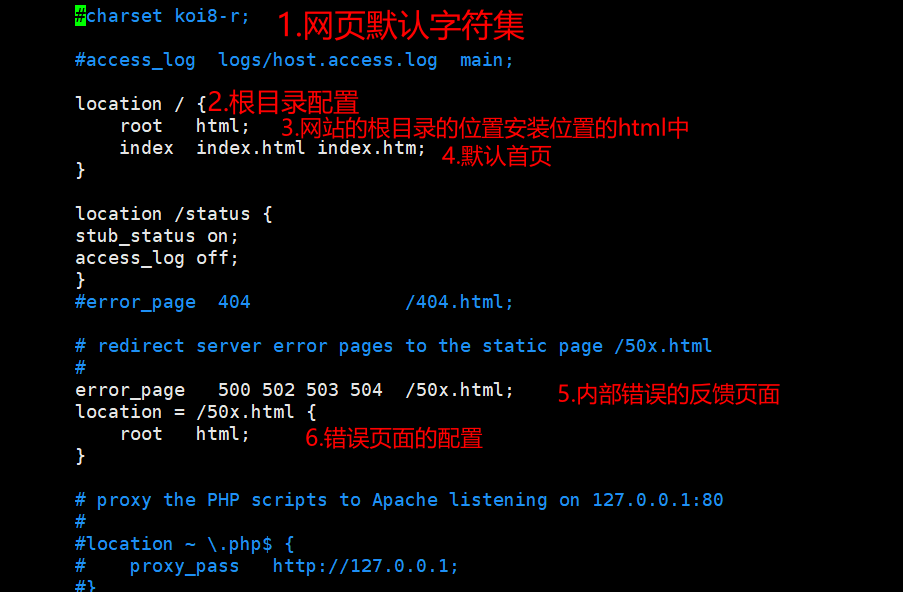
}



惊群现象：一个网路连接到来，多个睡眠的进程被同时叫醒，但只有一个进程能获得链接，这样会影响系统性能。

### 3.http配置（网页目录）





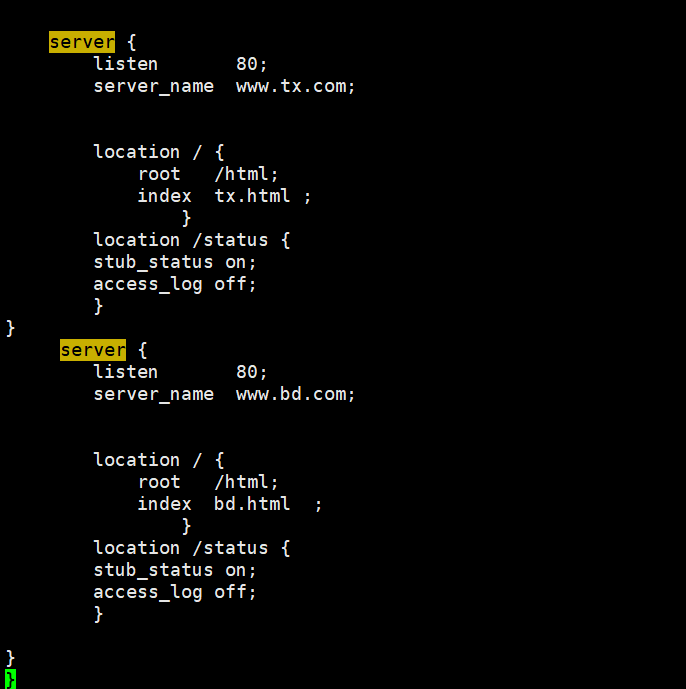
### 其它：

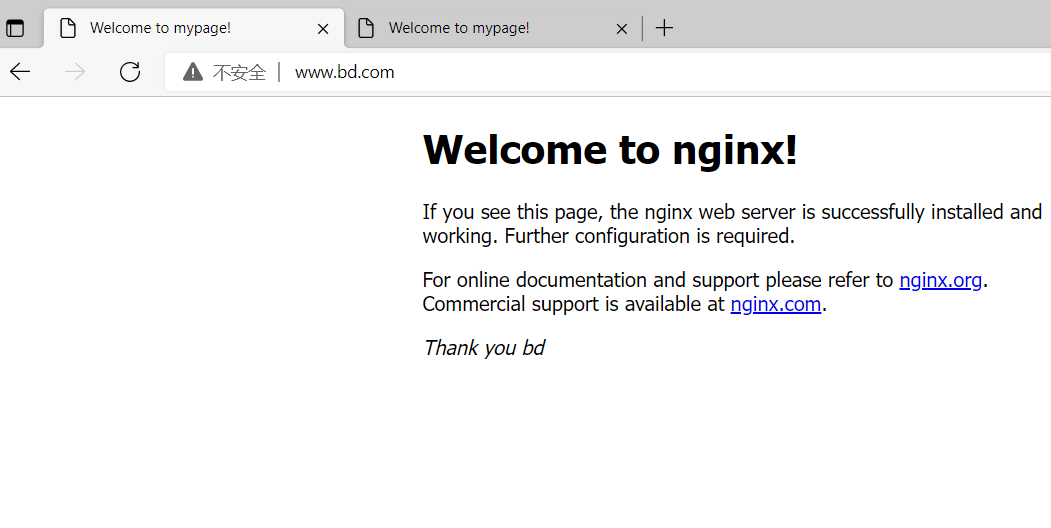


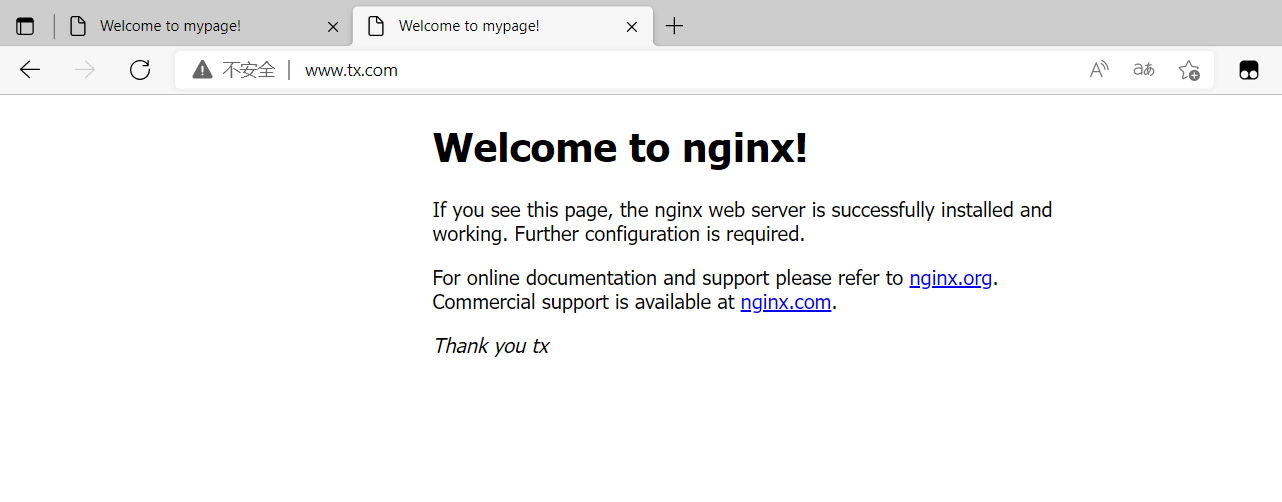
# 部署lnmp平台

修改nginx配置

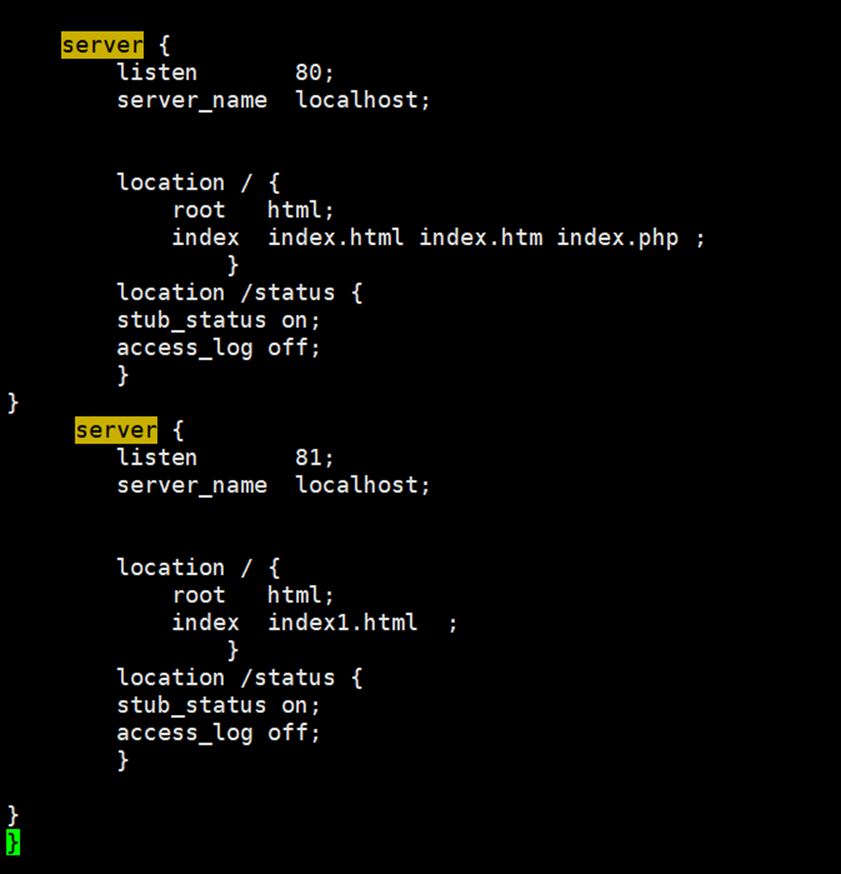
1.通过不同的网址进行访问不同的网页

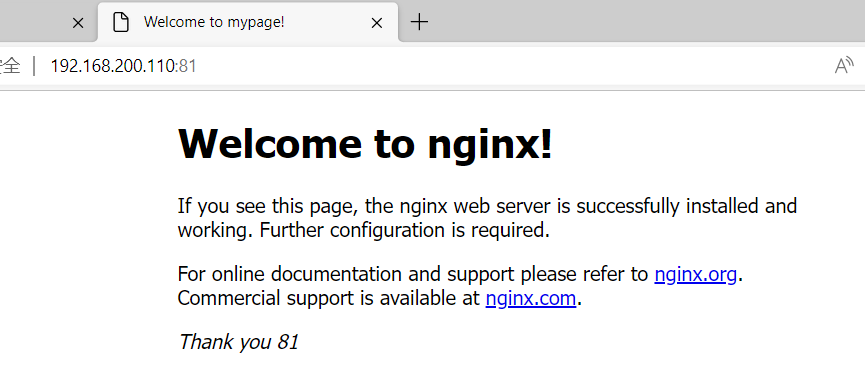


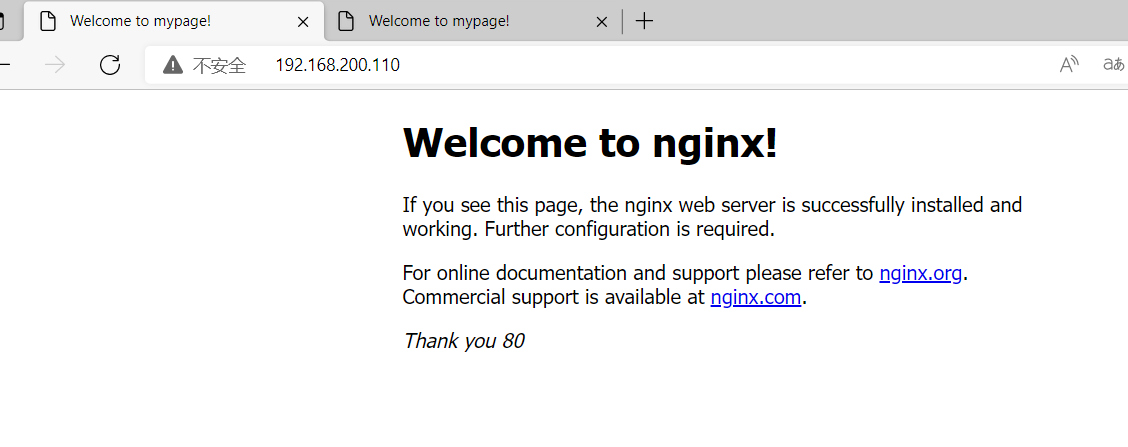




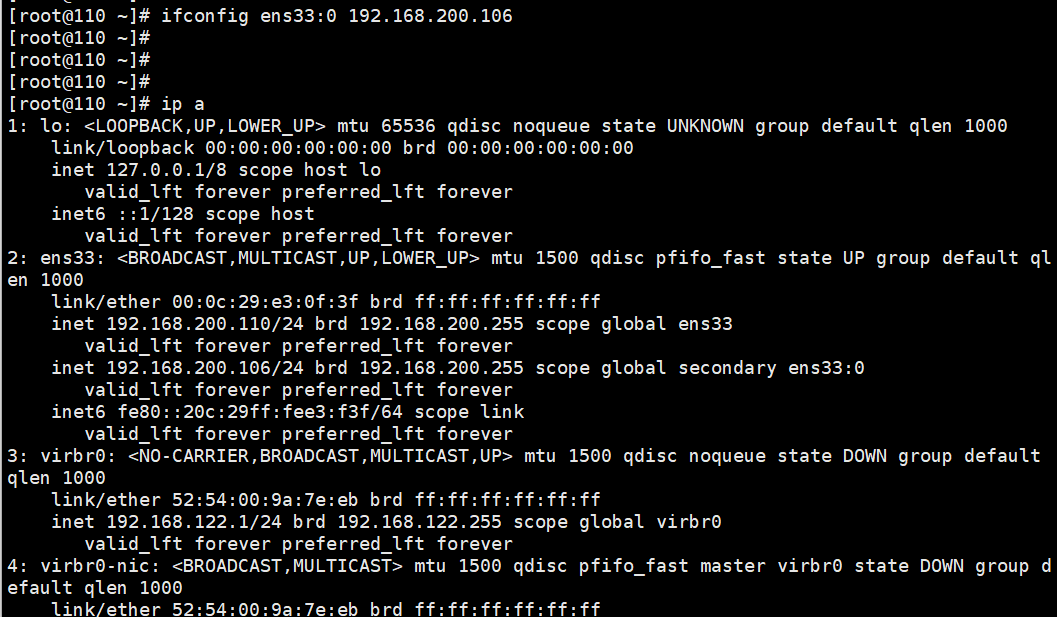
2.通过修改不同的端口访问不同的网页

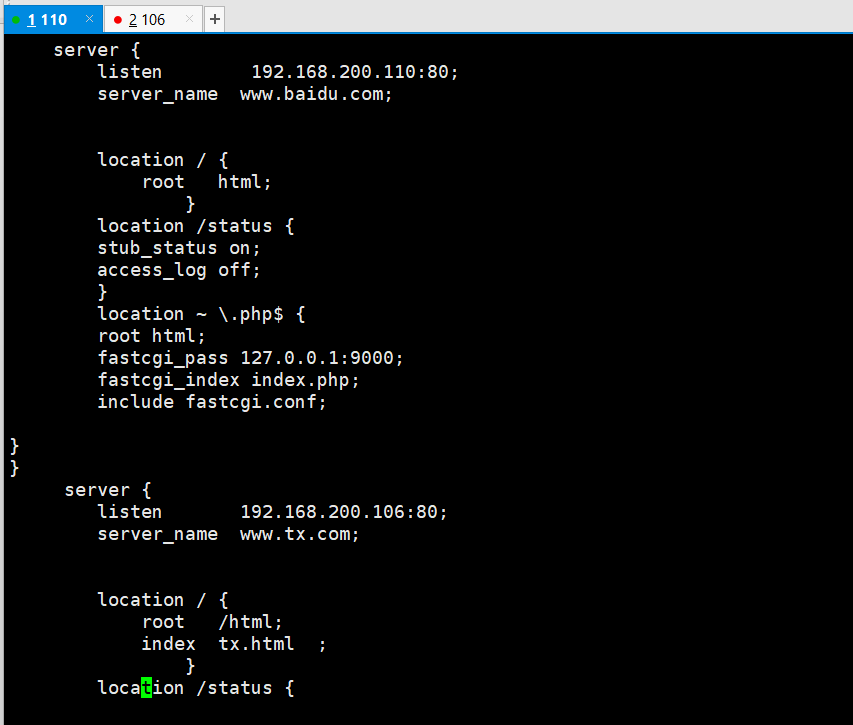


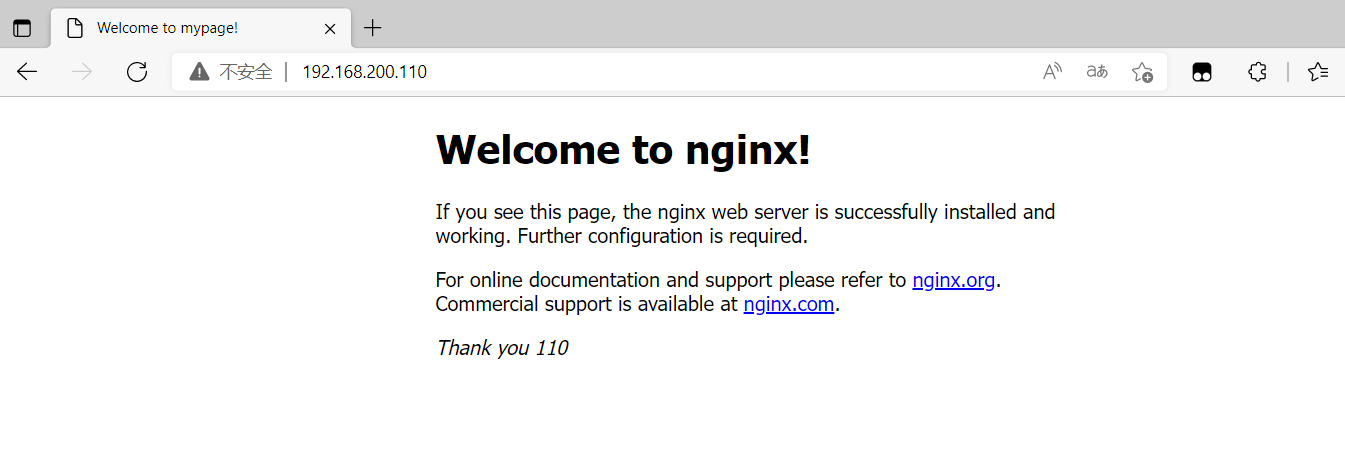


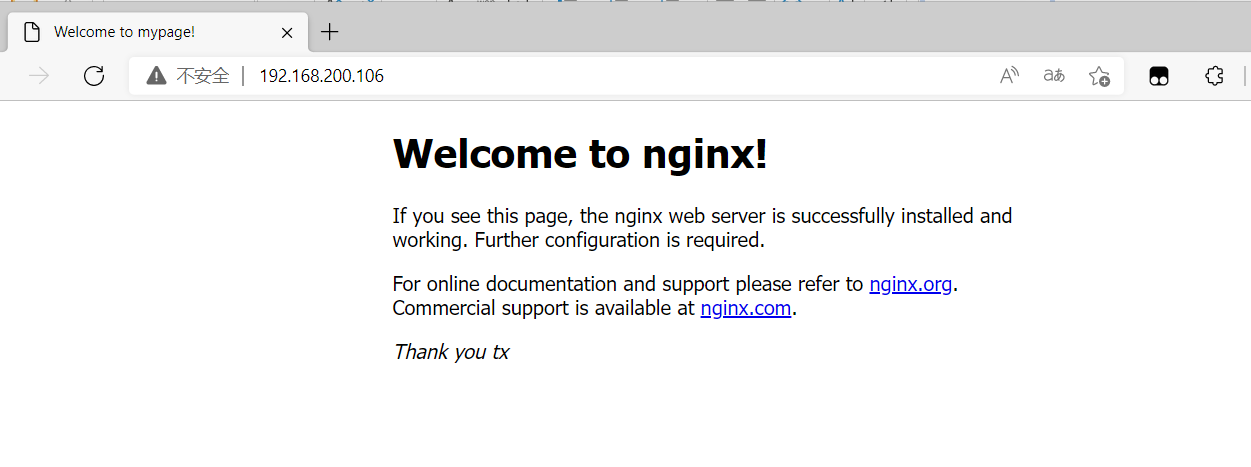


3.通过修改不同的IP地址访问不同的网页









## mysql安装

### 1.安装依赖

yum -y install ncurses-devel

yum -y install cmake

2.解压MySQL到usr/local/下，进入安装目录，配置，编译，安装

-DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql 数据库程序安装目录

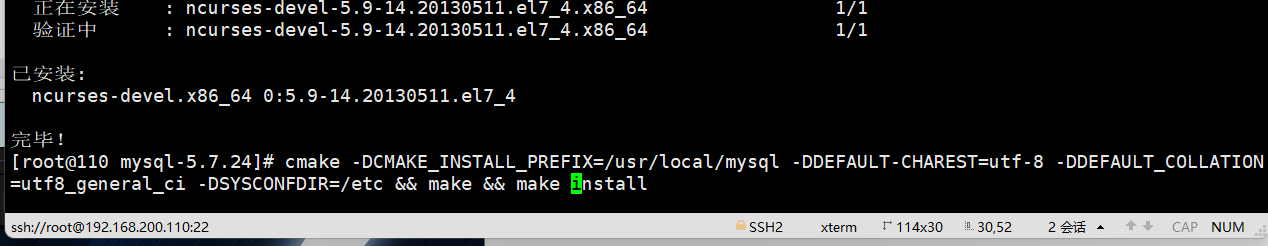
-DDEFAULT-CHAREST=utf-8 指定字符集编码

-DDEFAULT\_COLLATION=utf8\_general\_ci 默认的字符集校对规则utf8\_general\_ci

适用于utf-8字符集的通用规则

-DWITH\_EXTRA\_CHARESTS=all指定额外支持的字符集编码

-DSYSCONFDIR=/etc 指定配置文件存放目录



cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DDEFAULT\_CHARSET=utf8 -DDEFAULT\_COLLATION=utf8\_general\_ci -DWITH\_EXTRA\_CHARSETS=all -DSYSCONFDIR=/etc -DWITH\_BOOST=/usr/local/boost && make && make install

解决boost缺少问题



-DWITH\_BOOST=/usr/local/boost

### 2.安装后调整

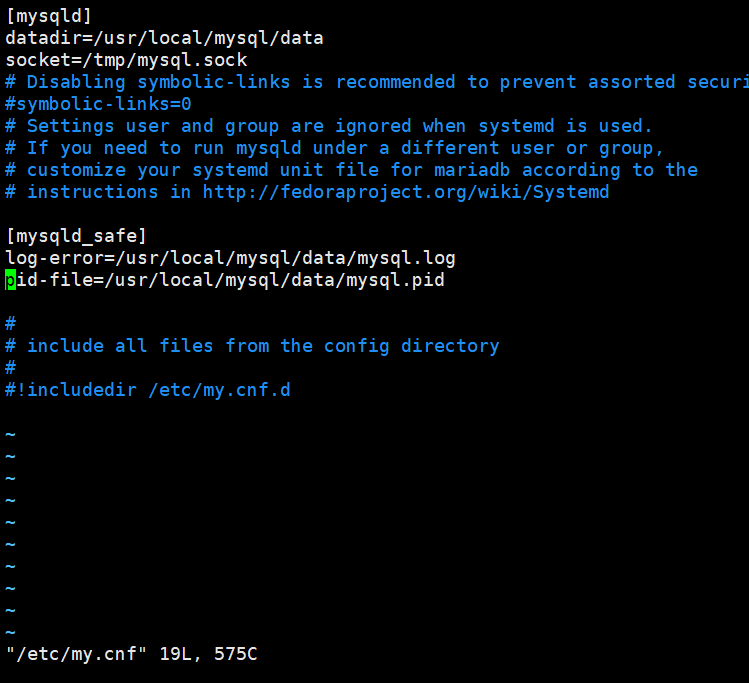
对数据库目录进行权限设置

cd /usr/local/mysql

chown -R mysql:mysql ./\*

建立配置文件（Centos默认支持mariadb数据库，系统默认的/etc/my.conf配置文件是mariadb的配置文件）

vim /etc/my.cnf



### 3.初始化数据库

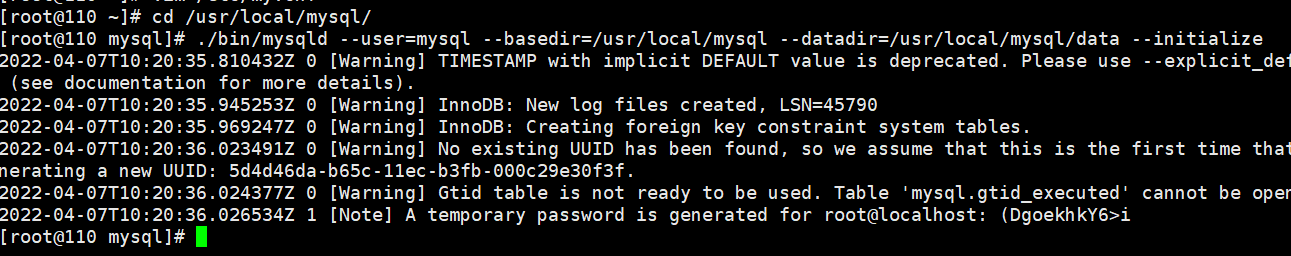
进到安装目录下

./bin/mysqld --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data --initialize

--basedir=/usr/local/mysql 指定安装目录

--datadir=/usr/local/mysql/data 指定数据目录

--user=mysql 指定用户身份



(DgoekhkY6>i

### 4.设置环境变量

echo "PATH=$PATH:/usr/local/mysql/bin" >> /etc/profile

./etc/profile = source /etc/profile

### 5.将mysql添加到系统服务

cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld

chmod +x /etc/init.d/mysqld

/etc/init.d/mysqld start

### 6.修改默认数据库密码

mysqladmin -u root -p'(DgoekhkY6>i' password '123456'

### 7.登录

mysql -uroot -p123456

## 安装php解析环境

php新版已经自带FPM（fast CGI process manager 进程管理器）连接nginx,对php解析实例进行管理。优化解析效率。因此在配置php编译选项时应添加“enable-fpm”以启用此模块

### 1.安装依赖

yum -y install libjpeg-devel libpng-devel libxml2-devel gd

### 2.进行配置

./configure --prefix=/usr/local/php5 --with-gd --with-mysql=/usr/local/mysql --with-mysqli=/usr/local/mysql/bin/mysql\_config --with-config-file-path=/usr/local/php5 --enable-mbstring --enable-fpm --with-jpeg-dir=/usr/lib && make && make install

### 3.安装后进行调整

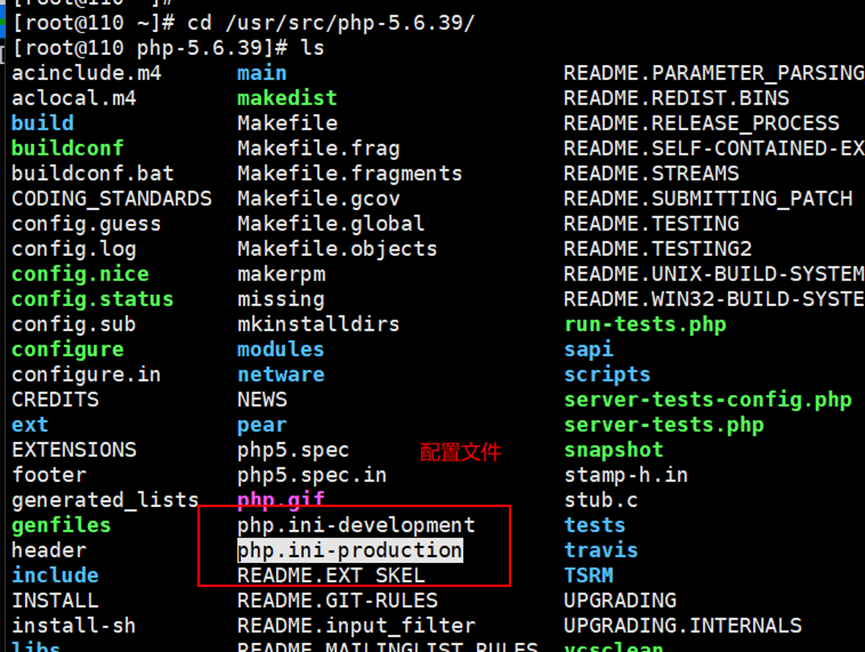
ln -s /usr/local/php5/bin/\* /usr/local/bin/

ln -s /usr/local/php5/sbin/\* /usr/local/sbin/

### 4.安装zendguardloader（提高php解析速率）

1.将安装包里面的配置文件复制到安装包里，拥有PHP的配置文件

cp php.ini-production /usr/local/php5/php.ini



2.进入zend解压目录下，把链接库文件移动到php下

cp ZendGuardLoader.so /usr/local/php5/lib/php/

3.修改配置文件，追加启动链接库

vim /usr/local/php5/php.ini

zend\_extension=/usr/local/php5/lib/php/ZendGuardLoader.so

zend\_loader.enable=1

4.进入php的etc下改名fpm的配置文件

mv php-fpm.conf.default php-fpm.conf

### 5.启动fpm（FastCGI Process Manager）,配置nginx并进行测试

若要让nginx能够解析php网页，有两种方法可以选择

1.nginx充当中介，将访问的php页面的web请求转交给其它服务器lamp去处理；

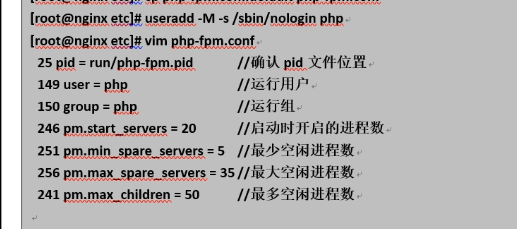
2.通过使用php的fpm模块来调用本机的php环境

如果使用fpm方式，启动php-fpm的进程，以便监听php解析请求。参考返例建立php-fpm.conf配置文件，修改pid文件、运行用户、服务数等相关设置，然后启动php-fpm程序。默认监听本机的9000端口

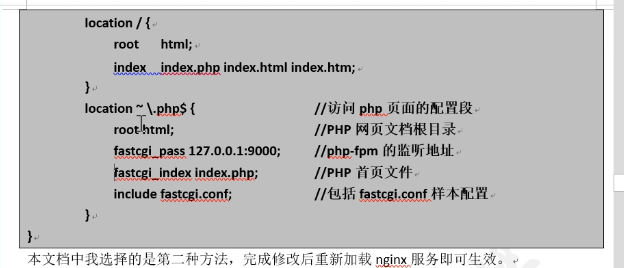
(1) useradd -M -s /sbin/nologin php

(2)cp /usr/local/php5/etc/php-fpm.conf /usr/local/php5/etc/php-fpm.conf.bak

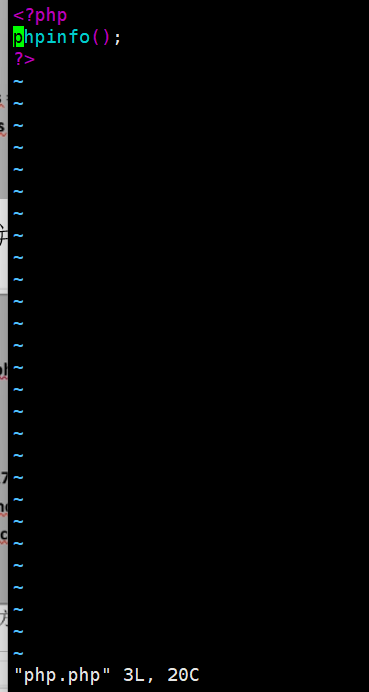
(3)vim /usr/local/php5/etc/php-fpm.conf

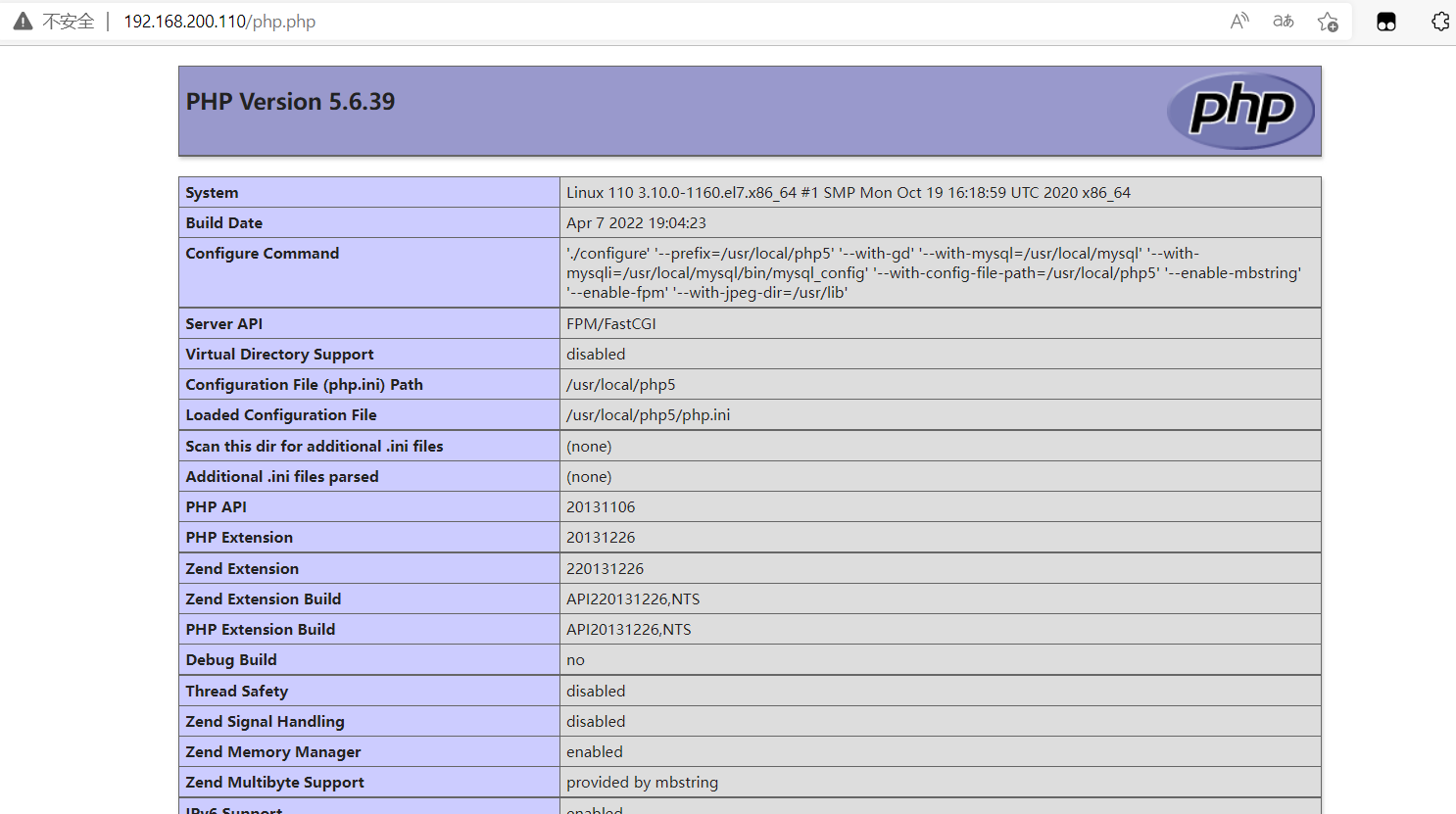


nginx配置，启动并测试

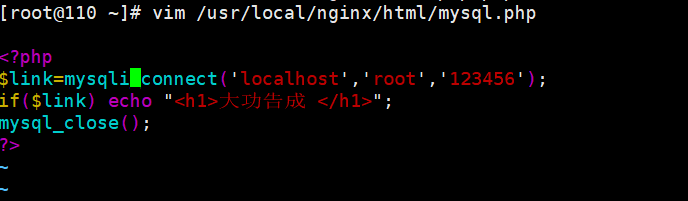


在html下创建PHP首页测试文件





mysql连接测试



<?php

$link=mysqli\_connect('localhost','root','123456');

if($link) echo "<h1>大功告成 </h1>";

mysqli\_close();

?>