Nginx反向代理与负载均衡高可用使用手册

[一、Nginx简介 2](#_Toc19552)

[二、 Centos上部署Nginx 3](#_Toc2350)

[1、 源码编译安装 3](#_Toc29408)

[2、 epel软件包安装 5](#_Toc5737)

[3、 yum标准包安装 5](#_Toc1829)

[三、 Nginx配置 6](#_Toc27481)

[1、 配置文件详解 6](#_Toc561)

[2、 主机与域名绑定 12](#_Toc32578)

[3、 反向代理配置 13](#_Toc24435)

[4、 负载均衡配置 14](#_Toc9251)

[四、 扩展包 15](#_Toc9208)

[1、 Lnmp一键安装包管理nginx 15](#_Toc441)

[2、 开源控制面板amh4.2管理nginx 17](#_Toc15332)

[五、 Nginx+keepalived双机热备部署 19](#_Toc4494)

[1、准备 19](#_Toc18626)

[2、部署 20](#_Toc27771)

[3、Keepalived的安装与配置 20](#_Toc3969)

[（2）备服务器配置 21](#_Toc32075)

[4、Rsync同步配置文件 22](#_Toc27404)

[（1） 主服务器配置Rsync 22](#_Toc2398)

[（2） 同步服务器脚本： 23](#_Toc4314)

# 一、Nginx简介

Nginx是一款[轻量级](http://baike.baidu.com/subview/1318763/16205192.htm" \t "http://baike.baidu.com/item/_blank)的[Web](http://baike.baidu.com/subview/3912/15992867.htm) 服务器/[反向代理](http://baike.baidu.com/view/1165595.htm)服务器及[电子邮件](http://baike.baidu.com/view/1524.htm)（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。由俄罗斯的程序设计师Igor Sysoev所开发，供俄国大型的入口网站及搜索引擎Rambler使用。其特点是占有内存少，[并发](http://baike.baidu.com/view/684757.htm" \t "http://baike.baidu.com/item/_blank)能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、[京东](http://baike.baidu.com/view/1412737.htm" \t "http://baike.baidu.com/item/_blank)、[新浪](http://baike.baidu.com/view/2410.htm)、[网易](http://baike.baidu.com/view/6043.htm)、[腾讯](http://baike.baidu.com/view/1591.htm)、[淘宝](http://baike.baidu.com/view/3629.htm)等。

nginx01

Nginx是现在最受欢迎的web服务器之一，它功能强大，消耗资源小，承受的并发量大是最受开发者喜欢的特性，所谓各有所长，在处理动态方面不如apache；在负载均衡方面，nginx是五层协议，而haproxy支持七层，LVS抗负载能力最强；但是nginx综合表现能力更受大家喜欢。

# Centos上部署Nginx

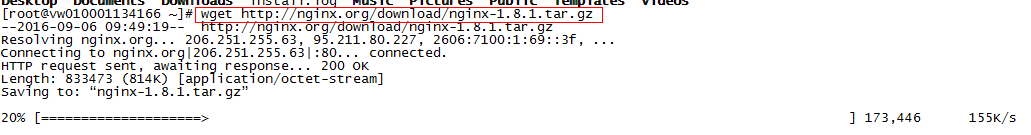
## 源码编译安装

首先在http://nginx.org/en/download.html 上下载nginx源码包

1.8.1下载地址：<http://nginx.org/download/nginx-1.8.1.tar.gz>

建议不要使用过高的版本，尽量使用stable version（稳定版）

（1）如图：使用wget 下载源码包



1. 解压并进入解压目录

tar xf nginx-1.8.1.tar.gz && cd nginx-1.8.1

1. 安装必要的包

yum -y install gcc gcc-c++ autoconf automake zlib zlib-devel openssl openssl-devel pcre-devel git

1. 创建nginx组和用户

groupadd -r nginx

useradd -s /sbin/nologin -g nginx -r nginx

1. 安装http\_substitutions\_filter\_module模块（反向代理替换模块）

git clone git://github.com/yaoweibin/ngx\_http\_substitutions\_filter\_module.git

1. 编译

./configure \

--prefix=/usr/local/nginx \

--sbin-path=/usr/local/nginx/sbin/nginx \

--conf-path=/usr/local/nginx/conf/nginx.conf \

--error-log-path=/usr/local/nginx/logs/error.log \

--pid-path=/usr/local/nginx/nginx.pid \

--user=nginx \

--group=nginx \

--with-http\_ssl\_module \

--with-http\_flv\_module \

--with-http\_gzip\_static\_module \

--http-log-path=/var/log/nginx/access.log \

--http-client-body-temp-path=/var/tmp/nginx/client \

--http-proxy-temp-path=/var/tmp/nginx/proxy \

--http-fastcgi-temp-path=/var/tmp/nginx/fcgi \

--with-http\_stub\_status\_module \

--add-module=/root/ngx\_http\_substitutions\_filter\_module

然后执行：make && make install

这样nginx编译成功，启动文件在/usr/local/nginx/sbin/nginx ，如何添加到服务和编写启动脚本这里就不详细介绍了.

## epel软件包安装

1. 安装中科大epel源

rpm -ivh \

http://mirrors.ustc.edu.cn/fedora/epel/6/x86\_64/epel-release-6-8.noarch.rpm

1. 安装nginx

yum -y install nginx

## 3、 yum标准包安装

1. 配置repo源

vi /etc/yum.repos.d/nginx.repo

插入：

[nginx]

name=nginx repo

baseurl=http://nginx.org/packages/centos/6/$basearch/

gpgcheck=0

enabled=1

1. 安装

yum -y install nginx

3种方式各取所需，编译安装虽然麻烦点，但是可以随心所欲的编译自己所需的模块。而epel软件包或者yum标准包安装都是定制的，但是后者非常的方便，如果需求不特别可以选择后者安装。

# Nginx配置

## 配置文件详解

user www www; #定义Nginx运行的用户和用户组  
worker\_processes 8; #nginx进程数，建议设置为等于CPU总核心数。  
error\_log /var/log/nginx/error.log info;

#全局错误日志定义类型，[ debug | info | notice | warn | error | crit ]

pid /var/run/nginx.pid; #进程文件  
worker\_rlimit\_nofile 65535; #建议与ulimit -n的值保持一致。  
events  
{  
use epoll; #参考事件模型，use [ kqueue | rtsig | epoll | /dev/poll | select | poll ];   
worker\_connections 65535; #单个进程最大连接数（最大连接数=连接数\*进程数）  
}  
http #设定http服务器  
{  
 include mime.types; #文件扩展名与文件类型映射表  
 default\_type application/octet-stream; #默认文件类型  
 #charset utf-8; #默认编码  
 server\_names\_hash\_bucket\_size 128; #服务器名字的hash表大小  
 client\_header\_buffer\_size 32k; #请求头部缓存  
 large\_client\_header\_buffers 4 64k; #设定请求缓存  
 client\_max\_body\_size 8m; #上传文件大小限制  
 sendfile on; #开启高效文件传输模式，sendfile指令指定nginx是否调用sendfile函数来输出文件，对于普通应用设为 on，如果用来进行下载等应用磁盘IO重负载应用，可设置为off，以平衡磁盘与网络I/O处理速度，降低系统的负载。注意：如果图片显示不正常把这个改成off。  
 autoindex on; #开启目录列表访问，合适下载服务器，默认关闭。  
 tcp\_nopush on; #防止网络阻塞  
 keepalive\_timeout 120; #长连接超时时间，单位是秒

#FastCGI相关参数是为了改善网站的性能：减少资源占用，提高访问速度。下面参数看字面意思都能理解。  
 fastcgi\_connect\_timeout 300;  
 fastcgi\_send\_timeout 300;  
 fastcgi\_read\_timeout 300;  
 fastcgi\_buffer\_size 64k;  
 fastcgi\_buffers 4 64k;  
 fastcgi\_busy\_buffers\_size 128k;  
 fastcgi\_temp\_file\_write\_size 128k;

#gzip模块设置  
 gzip on; #开启gzip压缩输出  
 gzip\_min\_length 1k; #最小压缩文件大小  
 gzip\_buffers 4 16k; #压缩缓冲区  
 gzip\_http\_version 1.0; #压缩版本（默认1.1，前端如果是squid2.5请使用1.0）  
 gzip\_comp\_level 2; #压缩等级  
 gzip\_types text/plain application/x-javascript text/css application/xml;  
 #压缩类型，默认就已经包含text/html，所以下面就不用再写了，写上去也不会有问题，但是会有一个warn。  
 gzip\_vary on;  
 #limit\_zone crawler $binary\_remote\_addr 10m; #开启限制IP连接数的时候使用

upstream www.gtafe.com {  
 #upstream的负载均衡，weight是权重，可以根据机器配置定义权重。weigth参数 表示权值，权值越高被分配到的几率越大。  
 [server](http://www.ha97.com/tag/server) 192.168.80.121:80 weight=3;  
 server 192.168.80.122:80 weight=2;  
 server 192.168.80.123:80 weight=3;  
 }

server #虚拟主机的配置  
{  
listen 80; #监听端口  
server\_name www.xxx.com xxx.com; #域名可以有多个，用空格隔开  
index index.html index.htm index.php;  
root /data/web/xxx;  
location ~ .\*\.(php|php5)?$  
{  
 fastcgi\_pass 127.0.0.1:8023;  
 fastcgi\_index index.php;  
 include fastcgi.conf;  
}  
#图片缓存时间设置  
location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$  
{  
 expires 10d;  
}  
#JS和CSS缓存时间设置  
location ~ .\*\.(js|css)?$  
{  
 expires 1h;  
}  
#日志格式设定  
log\_format access '$remote\_addr - $remote\_user [$time\_local] "$request" '  
'$status $body\_bytes\_sent "$http\_referer" '  
'"$http\_user\_agent" $http\_x\_forwarded\_for'; #定义本虚拟主机的访问日志  
access\_log /var/log/nginx/access.log access;

#对 "/" 启用反向代理  
location / {  
 proxy\_pass http://10.228.3.27:8023;  
 proxy\_redirect off;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 #后端的Web服务器可以通过X-Forwarded-For获取用户真实IP  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 #以下是一些反向代理的配置，可选。  
 proxy\_set\_header Host $host;  
 client\_max\_body\_size 10m; #允许客户端请求的最大单文件字节数  
 client\_body\_buffer\_size 128k; #缓冲区代理缓冲用户端请求的最大字节数，  
 proxy\_connect\_timeout 90; #nginx跟后端服务器连接超时时间(代理连接超时)  
 proxy\_send\_timeout 90; #后端服务器数据回传时间(代理发送超时)  
 proxy\_read\_timeout 90; #连接成功后，后端服务器响应时间(代理接收超时)  
 proxy\_buffer\_size 4k; #设置代理服务器（nginx）保存用户头信息的缓冲区大小  
 proxy\_buffers 4 32k; #proxy\_buffers缓冲区，网页平均在32k以下的设置  
 proxy\_busy\_buffers\_size 64k; #高负荷下缓冲大小（proxy\_buffers\*2）  
 proxy\_temp\_file\_write\_size 64k; #设定缓存文件夹大小，

}

#设定查看Nginx状态的地址  
location /NginxStatus {  
 stub\_status on;  
 access\_log on;  
 auth\_basic "NginxStatus";  
 auth\_basic\_user\_file conf/htpasswd;  
 #htpasswd文件的内容可以用apache提供的htpasswd工具来产生。  
}

#本地动静分离反向代理配置  
#所有jsp的页面均交由tomcat或[resin](http://www.ha97.com/tag/resin)处理  
location ~ .(jsp|jspx|do)?$ {  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 proxy\_pass http://127.0.0.1:8080;  
}

## 主机与域名绑定

在nginx的conf目录下面建立虚拟主机目录vhost

/usr/local/nginx/conf/vhost 这里用来存放主机配置文件

#如果是epel或者yum包安装的，可能配置文件放在/etc/nginx/conf.d

在nginx.conf 里面加入 include vhost/\*.conf

编写一个简单实例配置文件：xxx.com.conf

server

{

listen 80;

server\_name xxx.com www.xxx.com;

index index.html index.htm;

root /home/wwwroot/xxx.com;

access\_log /logs/xxx\_com.log;

error\_log /logs/xxx\_com\_error.log;

}

然后重载下nginx或者重启下nginx，配置就生效了，实现了把目录

/home/wwwroot/xxx.com 绑定到域名xxx.com上。

当然这里还有很多配置的参数，具体可以查看nginx文档。

文档地址：http://nginx.org/en/docs/

## 反向代理配置

在nginx的conf目录下面建立虚拟主机目录proxy

/usr/local/nginx/conf/proxy这里用来存放主机配置文件

在nginx.conf 里面加入 include proxy/\*.conf

编写一个简单实例配置文件：zabbix.v.xxx.com.conf

server {

listen 80;

server\_name zabbix.v.xxx.com;

index index.php index.html index.htm;

access\_log /logs/zabbix.log;

location / {

default\_type text/html;

proxy\_set\_header Referer http://10.228.3.70;

proxy\_set\_header Host 10.228.3.70;

proxy\_pass <http://10.228.3.70;> #这里是反代的地址

proxy\_set\_header Accept-Encoding "";

}

}

## 负载均衡配置

在vhost目录下面/usr/local/nginx/conf/vhost 创建quan.xxx.com.comf :

upstream quan.xxx.com {

ip\_hash;

server 10.228.2.47:8028 max\_fails=2 fail\_timeout=5s; #后面参数可选

server 10.228.3.27:8028 max\_fails=2 fail\_timeout=5s;

}

server {

listen 80;

server\_name quan.xxx.com ;

location / {

proxy\_pass http://quan.xxx.com;

}

}

upstream模块实现负载均衡，使用IP\_hash方式自动分配负载服务器，实现负载均衡。

Server 后面有个重要的参数weight（权重） ， 来处理2台服务器配置不同的情况

如47的配置比27的配置高一倍可以这样配置：

server 10.228.2.47:8028 weight=10;

server 10.2283.27:8028 weight=5;

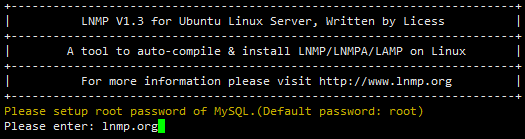
# 扩展包

# Lnmp一键安装包管理nginx

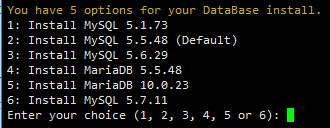
官网：<http://lnmp.org>

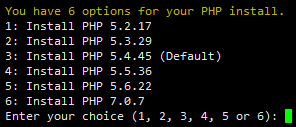
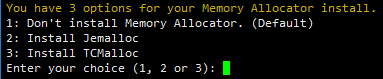
安装：**wget -c http://soft.vpser.net/lnmp/lnmp1.3-full.tar.gz && tar zxf lnmp1.3-full.tar.gz && cd lnmp1.3-full && ./install.sh** **lnmp**

按上述命令执行后，会出现如下提示：



需要设置MySQL的root密码。输入后回车进入下一步，如下图所示：  
 IMG_256

询问是否需要启用MySQL InnoDB，InnoDB引擎默认为开启，一般建议开启，直接回车或输入 y ，如果确定确实不需要该引擎可以输入 n，输入完成，回车进入下一步  
 

输入对应MySQL或MariaDB版本前面的序号，回车进入下一步，选择PHP版本：  
   
 输入要选择的PHP版本的序号，回车进入下一步，选择是否安装内存优化：  
   
 可以选择不安装、Jemalloc或TCmalloc，输入对应序号回车，直接回车为默认为不安装。

安装好之后，可以使用下面命令方便的管理nginx，主机域名设置，反带设置等等

**Usage: lnmp {start|stop|reload|restart|kill|status}**

**Usage: lnmp {nginx|mysql|mariadb|pureftpd|httpd} \**

**{start|stop|reload|restart|kill|status}**

**Usage: lnmp vhost {add|list|del}**

**Usage: lnmp database {add|list|edit|del}**

**Usage: lnmp ftp {add|list|edit|del}**

lnmp的优点：一键安装，可以选择版本，可以随意升级降级版本，随便编译模块，管理服务器方便，缺点：也不算缺点，没有可视化管理界面

# 开源控制面板amh4.2管理nginx

官网：http://amh.sh

安装：**wget http://amh.sh/file/AMH/4.2/amh.sh && chmod 775 amh.sh && ./amh.sh 2>&1 | tee amh.log**

**安装过程要求输入控制台密码和mysql密码。**

支持目前CentOS 6 x64、CentOS 6 i386、CentOS 5 x64、CentOS 5 i386版本

**安装完成之后输入http://IP:8888如图：**

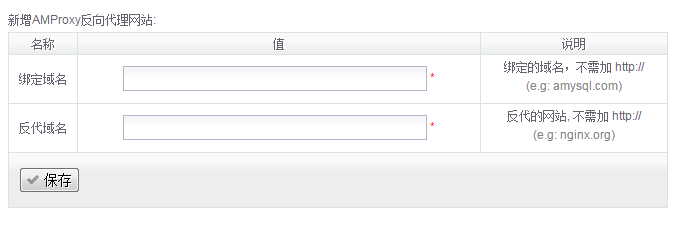


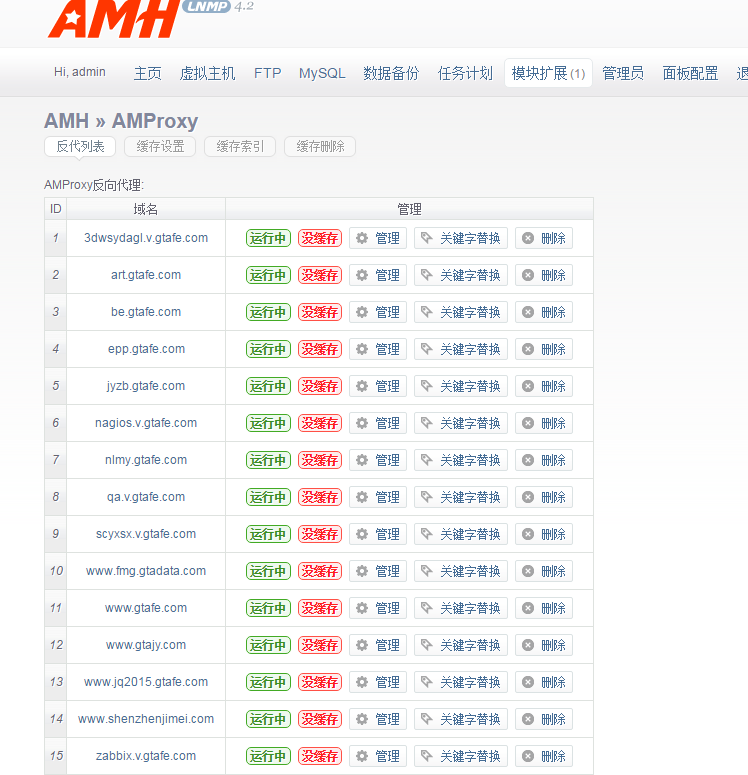
**登录之后：**



导航虚拟主机按钮可以管理nginx虚拟主机，在模块扩展中下载AMPoxy可以配置nginx反向代理。全可视化操作，解决了nginx配置难，不会配置的问题，当然他的配置相对简单，如果需求较高的话，还需要去安装目录进行手动配置

AMPoxy反带模块：





AMH面板的优势在于，容易操作，可视化面板，降低nginx学习成本。

# Nginx+keepalived双机热备部署

## 1、准备

Nginx服务器也有宕机的风险，所以使用keepalived构建VIP实现Nginx服务器的负载均衡，实现双机热备。

Nginx服务器（主）：10.228.3.26 （Centos6.5 X86\_64；4核/4G/50G）

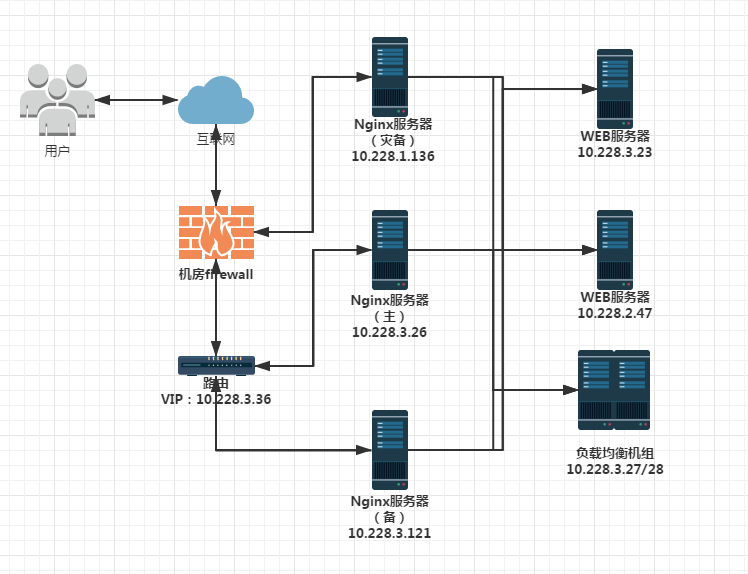
Nginx服务器（备）：10.228.3.121 （Centos6.5 X86\_64；4核/8G/50G）

Nginx服务器（灾备）：10.228.1.136 （Centos6.5 X86\_64；4核/8G/100G）

VIP : 10.228.3.36

公网IP：119.147.213.31 映射到VIP（10.228.3.36）

架构图:



## 部署

为了方便使用，3台都安装的扩展包AMH4.2（安装方法见第四部分：扩展包）

nginx目录在：/usr/local/nginx

nginx主配置文件在：/usr/local/nginx/conf/nginx.conf

虚拟主机配置文件：/usr/local/nginx/vhost

反向代理配置文件：/usr/local/nginx/proxy

#我们这里主要是反带为主，这里是主要配置文件存放的地方。

可视化后台管理地址：<http://IP:8888> 登录帐号密码：admin/password

Keepalived配置文件位置：/etc/keepalived/keepalived.conf

## Keepalived的安装与配置

在主备Nginx服务器上都安装keepalived,灾备不用安装，我们使用yum安装：

yum -y install keepalived

（1）主服务器配置

global\_defs {

router\_id Nginx\_Master

}

vrrp\_instance VI\_1 {

state MASTER #主用MASTER

interface eth0 #监测网络接口

virtual\_router\_id 50 #主、备必须一样

priority 100 #优先级：主＞备

advert\_int 1

authentication {

auth\_type PASS #VRRP认证，主备一致

auth\_pass 1111 #密码

}

virtual\_ipaddress {

10.228.3.36 #VRRP HA虚拟地址

}

}

### （2）备服务器配置

global\_defs {

router\_id Nginx\_Backup

}

vrrp\_instance VI\_1 {

state BACKUP #备用BACKUP

interface eth0 #监测网络接口

virtual\_router\_id 50 #主、备必须一样

priority 99 #优先级：主＞备

advert\_int 1

authentication {

auth\_type PASS #VRRP认证，主备一致

auth\_pass 1111 #密码

}

virtual\_ipaddress {

10.228.3.36 #VRRP HA虚拟地址

}

}

然后启动keepalived:

service keepalived start

加入开机启动：

chkconfig keepalived on

## Rsync同步配置文件

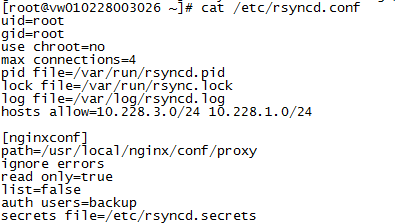
主备nginx服务器的配置应该是一致的，所以我们需要定时同步配置文件，我们这里使用Rsync进行同步。

### 主服务器配置Rsync

Centos已经已经默认安装了Rsync，所以我们跳过安装这一步骤。在Nginx主服务器（被同步服务器即：MASTER）上操作：

A、我们首先修改/etc/xinetd.d/rsync 文件，把disable改成no之后保存。

B、创建rsync配置文件：/etc/rsyncd.conf:插入下面的内容



C、创建配置文件中的密码文件/etc/rsyncd.secrets 输入：bacup:password

用户：密码 这样的格式，其中用户是配置文件中的auth users

D、启动Rsync：

rsync --daemon --config=/etc/rsyncd.conf

### 同步服务器脚本：

A、首先创建密码文件，我们这里创建在/root/sync/pass

插入主服务器用户BACKUP用户密码：password

B、编写同步脚本：

#!/bin/bash

syncdir="/usr/local/nginx/proxy/"

rsync -avzP --delete --password-file=/root/sync/pass \

[backup@10.228.3.26::nginxconf](mailto:backup@10.228.3.26::nginxconf) ${syncdir}

C、定时执行

crontab -e 中加入：

0 \*/1 \* \* \* /root/sync/rsync.sh >/root/sync/error.log 2>&1

每小时执行一次，错误输出到error.log文件中。

我们打开10.228.3.36 发现会打开26服务器上的网站，然后我们关闭26服务器，继续打开10.228.3.36，发现Nginx已经调整到121上了。这样子就实现了双机热备，实现了nginx服务器的高可用，当2台机器都宕机的情况下，我们还可以手动切换到灾备服务器。