**案例1：**

**克隆 4 台新虚拟机,IP 配置如下:**

虚拟机 7 eth2:201.1.1.100

虚拟机 8 eth2:201.1.1.5 eth3:201.1.2.5

虚拟机 9 eth3:201.1.2.100

虚拟机 10 eth3:201.1.2.200

a) 在虚拟机 7 上源码安装 Nginx 软件,要求如下:

b) 通过./configure --help 查看该软件支持的模块(--with开头)

c) 要求安装 ssl 模块、stream 模块以及 stub\_status 模块

d) 安装前创建用户 nginx,安装时指定安装用户为 nginx

e) 百度搜索 nginx 的 stub\_status 模块的作用,以及如何修改配置文件,通过 stub\_status 查看 nginx 服务器的状态

###配置一键安装Nginx脚本###

vim /root/1.sh

#!/bin/bash

tar -xvf lnmp\_soft.tar.gz

cd lnmp\_soft/

tar -xvf nginx-1.12.2.tar.gz

cd nginx-1.12.2/

useradd -s /sbin/nologin nginx

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

./configure --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module

make && make install

clear

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

ln -s /usr/local/nginx/conf/nginx.conf /nginx.conf.link

nginx -V

nginx

**案例2：**

**在虚拟机 7 上修改 nginx 配置文件,实现基于域名的虚拟主机:**

a) 域名分别为 www.tarena.com 和 www.tedu.cn

b) 注意客户端验证需要修改/etc/hosts

c) 访问 www.tedu.cn 时会提示输入用户名与密码,输入正确才可以显示流量页面

d) 页面内容任意

###服务端配置基于域名的虚拟主机###

vim /nginx.conf.link

server {

listen 80;

server\_name www.tarena.com;

charset utf-8;

location / {

root html;

index index.html index.htm;

}

}

server {

listen 80;

server\_name www.tedu.cn;

auth\_basic "Input passwd:";

auth\_basic\_user\_file "/usr/local/nginx/pass";

location / {

stub\_status on ;

allow 201.1.1.5;

allow 127.0.0.1;

deny all;

}

}

yum -y install httpd-tools

htpasswd -c /usr/local/nginx/pass tom

htpasswd /usr/local/nginx/pass jerry

echo “这是TARENA.COM 河马。” > /usr/local/nginx/html/index.html

nginx -s reload

###客户端验证域名内容###

echo “201.1.1.100 [www.tarena.com](http://www.tarena.com) [www.tedu.cn](http://www.tedu.cn)” >> /etc/hosts

firefox [http://www.terena.com](http://terena.com)

firefox http://www.tedu.cn

**案例3：**

**虚拟机 8,9,10 均安装 nginx 软件作为服务器**

a) 虚拟机 7 作为客户端

b) 虚拟机 8 的 nginx 配置为集群调度器,算法为轮询算法,集群池为 201.1.2.100 和 201.1.2.200,配置最大失败次数为 1,失败超时时间为 30 秒,201.1.2.100 的权重为 2

c)虚拟机 9 和虚拟机 10 配置为 LNMP 环境,支持 PHP 动态页面

d)在虚拟机 9 上创建测试页面(静态页面至少一个,连接数据库的动态页面至少一个,参考lnmp\_soft/php\_scripts/mysql.php)

e)配置 rsync 自动将虚拟机 9 上的页面自动同步到虚拟机 10,确保两台主机的页面一致

###编写虚拟机9-10一键搭建LNMP环境脚本###

虚拟机9、10：

vim /root/2.sh

#!/bin/bash

yum -y install mariadb-server mariadb mariadb-devel

yum -y install php php-mysql

cd /root/lnmp\_soft/

yum -y install php-fpm-5.4.16-42.el7.x86\_64.rpm

systemctl start mariadb php-fpm

systemctl enable mariadb php-fpm

cd /root/lnmp\_soft/php\_scripts/

cp mysql.php /usr/local/nginx/html/

###编写虚拟9实时同步脚本###

tar -xvf inotify-tools-3.13.tar.gz

cd inotify-tools-3.13/

./configure && make && make install

vim /root/3.sh

#!/bin/bash

while inotifywait -rq /usr/local/nginx/ &>> /inoti.log

do

rsync -az --delete /usr/local/nginx/ root@201.1.2.200:/usr/local/nginx/ &>>/rsync.log

done

/root/3.sh &

###虚拟机9配置动静分离###

vim /nginx.conf.link

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi\_params;

}

###虚拟机8搭建HTTP调度###

vim /nginx.conf.link

http {

upstream webs {

server 201.1.2.100 weight=2 max\_fails=1 fail\_timeout=30;

server 201.1.2.200 weight=1 max\_fails=1 fail\_timeout=30;

}

server {

location / {

proxy\_pass http://webs;

}

}

}

###虚拟机7测试反向代理效果###

firefox 201.1.1.5/mysql.php #刷新3次

**案例4：**

**延续前面的实验,优化所有的 nginx 软件,要求如下:**

a) 所有 nignx 均不要显示版本号

b) 并发连接数调整为 65536

c) 调整后使用 ab 软件测试并发量

d) 使用“watch -n 1 ss -s”命令观察服务器连接数变化(每秒执行一次 ss 命令)

e)百度搜索 ss -s 命令的用法

###取消nginx版本号显示###

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

server {

server\_tokens off;

...

}

###提高并发量并测试###

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

events {

worker\_connections 65535;

}

ulimit -Hn 100000

ulimit -Sn 100000

vim /etc/security/limits.conf #重启即生效

\* soft nofile 100000

\* hard nofile 100000

ab -c2000 -n2000 http://127.0.0.1/

**Evening：**

1. rsyslog 系统日志守护进程

2、系统日志配置文件：

/etc/rsyslog.d/listen.conf

/etc/rsyslog.conf #定义Linux所有日志的位置

规则（选择器+动作）

选择器：设施（种类）+级别（7个等级） 用.号分隔

3、日志设置样例：

cron.\* /var/log/cron

#将cron守护进程所有信息记录在cron中

mail.warn /var/log/mail.warn

#将mail守护进程所有高于等于warn的消息都记录到mail.warn中

mail.=info /var/log/mail.info #=号强制记录单一级别

mail.!info /var/log/mail.info #！排除这类信息

none #关闭日志

logger -p lpr.info -t ooxx ‘xixihaha’

#写日志命令，-p（日志种类+等级），-t（标题），’’（内容）

1. 实验：

vim /etc/rsyslog.d/listen.conf

local0.inflocal0 /var/log/info1.log

local0=info /var/log/info0.log

local0!info /var/log/info2.log

logger -p local0.info -t abc ‘test111’ #查看info0.log、info1.log

logger -p local0.debug -t abc ‘test222’#查看info2.log

mail.\* -/var/log/maillog #-号为异步日志，有可能会丢失

虚拟机1：

#$ModLoad imudp

#$UDPServerRun 514 #把UDP监听端口打开

#$ModLoad imtcp

#$InputTCPServerRun 514 #把TCP监听端口打开

local2.notice /var/log/test1.log #设置日志接收位置

虚拟机2：

local2.notice @192.168.1.15:514 #定义UDP网络日志

local2.notice @@192.168.1.15:514 #定义TCP网络日志

logger -p local2.notice -t abc test12345

