1.克隆4台虚拟机，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:

安装依赖软件包：Yum -y install gcc openssl-devel pcre-devel zlib-devel

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

开头) ./configure --help

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

进入./configure模式，按要求配置：

./configure --with--http\_ssl\_module --with-http\_stub\_status\_module

d) 安装前创建用户 nginx,安装时指定安装用户为 nginx

创建用户：Useradd -s /sbin/bash nginx

指定安装用户：./configure ---user=nginx --group=nginx

e) 百度搜索 nginx 的 stusb\_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

进入配置文件，进行更改:vim /usr/local/nginx/conf.nginx.conf

http {

......

Sendfile on;

Tcp\_nopush on;

Keepalive-timeout 65;

Tcp\_nodelay on;

Server\_tokens off;

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,输入正确才可以流量页面

d) 页面内容任意

A c :修改配置文件：vim /usr/local/nginx/conf/nginx.conf:

Server {

Listen 80;

Server\_name [www.tarena.com](http://www.tarena.com/);

Auth\_basic “Input password”;

Auth\_basic\_user\_file “/usr/local/nginx/pass”;

Location / {

Root html;

Nginx index.html index.htm;

}

}

.....

Server {

Listen 80 ;

Server\_name www.tedu.cn;

Location / {

Root www;

Index index.html index.htm;

}

}

B : 修改配置文件：vim /etc/hosts

.....

201.1.1.5 [www.tarena.com](http://www.tarena.com/) [www.tedu.cn](http://www.tedu.cn/)

D : firefox [http://www.tarena.com](http://www.tarena.com/)

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

修改配置文件：vim /usr/local/nginx/conf/nginx.conf

....

http {

....

Upstream webserver {

Ip\_hash;

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;

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

下载相关依赖包:gcc openssl-devel pcre-devel zlib-devel php php-mysql php-fpm nginx(需编辑并加载安装) mariadb mariadb-server mariadb-devel

停止httpd服务，重启php-fpm和mariadb服务.

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)

修改配置文件:Vim /usr/local/nginx/conf/nginx.conf

Location / {

Root html

Index index.php index.html index.htm;

}

Location ~ \.php$ {

Root html;

Fastcgi\_pass 127.0.0.1:9000;

Fastcgi\_index index.php;

Include fastcgi.conf;

}

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

....

Upstream webs {

Server 201.1.2.100:80;

Server 201.1.2.200:80;

}

Server {

Listen 80;

Server\_name localhost;

Localhost / {

Proxy\_pass [http://webs](http://webs/);

Root html;

Index index.html index.htm;

}

}

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(百度搜索)

修改配置文件:vim /usr/local/nginx/conf/nginx.conf

http {

....

Sendfile on;

Tcp\_nopush on;

Keepalive\_timeout 65;

Tcp\_nodelay on;

Server\_tokens off;

....

}

更改php配置文件:vim /etc/php-fpm.d/www.conf

Fastcgi\_param server\_software nginx;

b) 并发连接数调整为 65536

更改配置文件:vim /usr/local/nginx/conf/nginx.conf

....

Worker \_processes 2;

Events {

Worker \_connections 65535;

Use epoll;

}

....

c) 调整后使用 ab 软件测试并发量

Ab -n 2000 -c 2000 http://201.1.1.5/

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

5. 附加题目:优化 Linux 内核参数,要求如下:

a) 百度搜索:tcp\_syncookies 参数、用法、如果调整值

参数：net.ipv4.tcpsyncookies = 1

#表示开启重用。允许将TIME-WAIT sockets重新用于新的TCP连接，默认为0，表示关闭；

b) 百度搜索:tcp\_fin\_timeout 参数、用法、如果调整值

net.ipv4.tcpfintimeout = 30

#表示如果套接字由本端要求关闭，这个参数决定了它保持在FIN-WAIT-2状态的时间。

c) 百度搜索:tcp\_retries 参数、用法、如果调整值

6. 在虚拟机 9 和 10 上部署 PHP 登陆环境:

a) 部署 lnmp 环境(支持动静分离)

更改配置文件:vim /usr/local/nginx/conf/nginx.conf

Location / {

Root html;

Index index.php index.html index.htm

}

Location ~ \.php$ {

Root html;

fastcgi\_pass 127.0.0.1:9000;

Fastcgi\_index index.php;

Include fastcgi.php;

}

b) 部署测试性页面(参考 lnmp\_soft/php\_scripts/php-memcached-demo.tar.gz

解压软件包:tar -xf php-memcached-demo.tar.gz

复制php-memcached-demo.tar.gz/文件内的所有内容到/usr/local/nginx/html

访问用户，填写信息:firefox http://201.1.2.100

查看访问信息：ls /var/lib/php/session

c) 测试登陆页面的效果

解压php-memcached-demo.tar.gz:tar -xf php-memcached-demo.tar.gz

将该软件包所有文件目录复制到/usr/local/nginx/html下

d) 查看/etc/php-fpm.d/www.conf 配置文件,确认 session 会话信息保存位置

/var/lib/php/session

e) 到/var/lib/php/session 目录下查看 session 文件

7. 使用 memcached 实现会话信息共享

a) 虚拟机 8 安装部署 memcached 缓存数据库

先安装软件包：Yum -y install memcached

更改配置文件：vim /usr/lib/systemd/system/memcached.service

Execstart=/usr/lib/memcached -u $USER -p $PORT -m $CACHESIZE -c $MAXCONN $OPTIONS

Vim /etc/sysconfig/memcached

PORT=”11211”

USER=”memcached”

MAXCONN=”1024”

CACHESIZE=”64”

OPTIONS=””

b) 修改虚拟机 9 和 10 的配置,让 PHP 保存会话信息到 memcached

修改配置文件：vim /etc/php-fpm.d/www.conf

Php\_value [session.save\_handler] = memcache

Php\_value [session.save\_path] = “tcp://201.1.1.5:11211”

c) 默认 PHP 无法连接 memcached,需要安装对应的扩展模块

安装php-pecl-memcache软件包，并启动：

yun -y install php-pecl-memcache

systemctl restart php-fpm

d) 完成后再次通过集群调度器访问登陆页面,测试登陆效果(为了查看效果,可用将后台页面的背景颜色适当调整)

先解压压缩包:tar -xf /root/lnmp\_soft/php-scripts/php-memcacha-demo.tar.gz

进入php解压后文件：ls /root/lnmp\_soft/php-scripts/php-memcacha-demo/

将php文件下所有目录文件复制到网络文件下：

cp -r php-memcacha-demo/\* /usr/local/nginx/html

更改配置文件：vim /etc/php-fpm.d/www.conf