

# 作业

## 一、访问lvs服务

### 1.环境配置



宿主机cip: 192.168.43.242

Lvs: vip: 192.168.43.128

dip: 192.168.241.131

RS1: rip1:192.168.241.132

gateway: 192.168.241.131

RS2: rip2:192.168.241.133

gateway: 192.168.241.131

### 2.ipvsadm搭建集群，部署方式采用NAT模式：

```
1 ipvsadm -A -t 192.168.43.128:80 -s rr //以轮询方式创建集群
2 //查看
3 ipvsadm -Ln
4
5 ipvsadm -a -t 192.168.43.128:80 -r 192.168.241.132 -m //-m表示NAT模式
6 ipvsadm -a -t 192.168.43.128:80 -r 192.168.241.133 -m
```

```
TCP 192.168.43.128:80 rr
```

```
-> 192.168.241.132:80 Masq 1 0 0
```

```
-> 192.168.241.133:80 Masq 1 0 0
```

### 3.客户端curl vip结果：

```
C:\Users\21499>curl 192.168.43.128
hello world(rs1)

C:\Users\21499>curl 192.168.43.128
hello world(rs2)

C:\Users\21499>curl 192.168.43.128
hello world(rs1)

C:\Users\21499>curl 192.168.43.128
hello world(rs2)
```

### 3.1 access\_log 显示是 curl 进行的访问:

```
192.168.43.242 - - [10/Jun/2024:23:57:11 -0700] "GET /hello.html HTTP/1.1" 200 16 "-" "curl/8.4.0" "-"
192.168.43.242 - - [10/Jun/2024:23:57:14 -0700] "GET /hello.html HTTP/1.1" 200 16 "-" "curl/8.4.0" "-"
192.168.43.242 - - [10/Jun/2024:23:59:13 -0700] "GET /mi.html HTTP/1.1" 200 18 "-" "curl/8.4.0" "-"
192.168.43.242 - - [10/Jun/2024:23:59:31 -0700] "GET /mi.html HTTP/1.1" 200 18 "-"
```

## 4. 本地 hosts 文件加域名映射 /etc/hosts

```
# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost
192.168.43.128 www.example.com
```

## 5. 分别完成两台 nginx 的静态资源部署

### 5.1 在两台 nginx 机器的 server 块中添加 location 块

```
1     location /hello.html{
2         root    /usr/share/nginx/html;
3         index  hello.html hello.htm;
4     }
5     location /mi.html{
6         root    /usr/share/nginx/html;
7         index  mi.html mi.htm;
8     }
```

### 5.2 添加 hello.html 静态资源:

```
1 #RS1 添加如下内容:
2 hello world(rs1)
3 #RS2 添加如下内容:
4 hello world(rs2)
```

### 5.3添加 mi.html 静态资源:

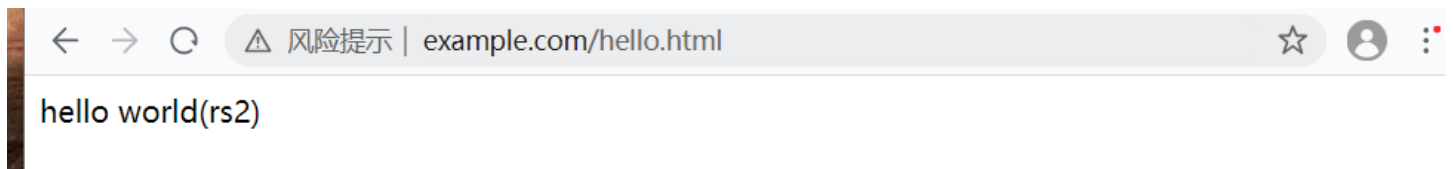
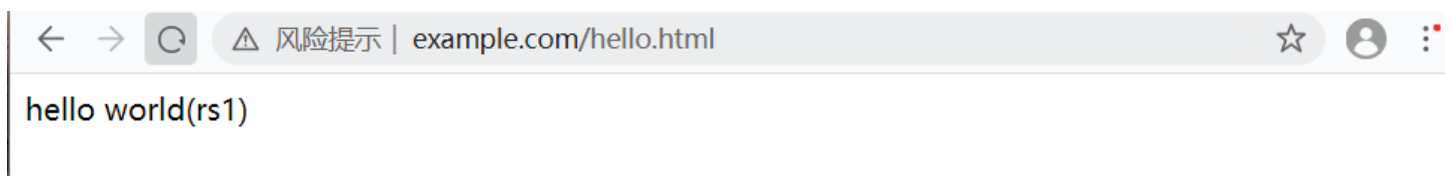
```
1 #RS1 添加如下内容:
2 i love xiaomi(rs1)
3 #RS2 添加如下内容:
4 i love xiaomi(rs2)
```

### 6.通过域名去访问页面

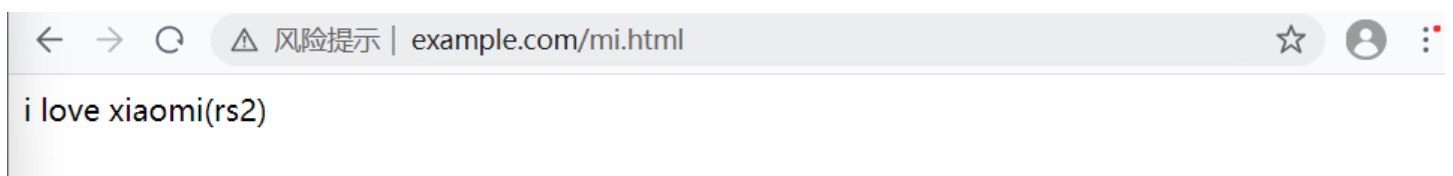
```
C:\Users\21499>curl www.example.com/hello.html
hello world(rs1)
C:\Users\21499>curl www.example.com/hello.html
hello world(rs2)
C:\Users\21499>curl www.example.com/mi.html
i love xiaomi(rs1)
C:\Users\21499>curl www.example.com/mi.html
i love xiaomi(rs2)
```

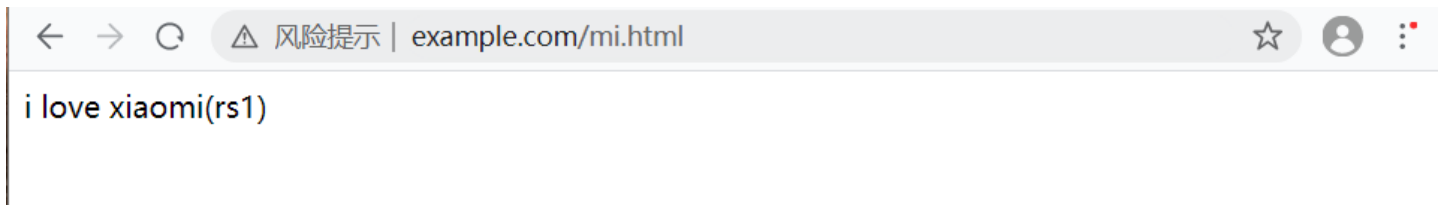
### 7.通过浏览器去访问

访问hello.html页面



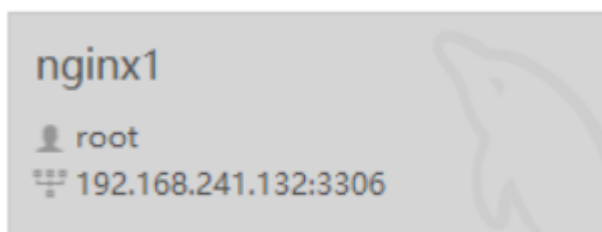
访问mi.html页面





## 二、通过脚本分析nginx机器上的access 日志

### 1.连接数据库



### 2.创建数据库

2.1先通过 `mysql -u root -p` 命令进入数据库。

```
[root@localhost ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.37 MySQL Community Server - GPL
```

#### 2.2 创建数据库

使用 `CREATE DATABASE nginx_logs;` 命令创建数据库。

```
mysql> CREATE DATABASE nginx_logs;
Query OK, 1 row affected (0.05 sec)
```

#### 2.3选择数据库

使用 `USE nginx_logs;` 命令选择数据库。

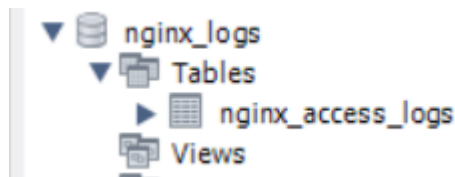
```
mysql> USE nginx_logs;
Database changed
```

#### 2.4创建表

```
mysql> CREATE TABLE nginx_access_logs (  
-> id INT AUTO_INCREMENT PRIMARY KEY,  
-> timestamp DATETIME NOT NULL,  
-> qps FLOAT NOT NULL,  
-> http_200 INT NOT NULL,  
-> http_500 INT NOT NULL  
-> );  
Query OK, 0 rows affected (0.12 sec)
```

## 2.5查看是否创建成功

数据库中存在对应的数据库和表。



## 3.编写shell脚本

### 3.1 进入 /usr/local/bin目录下创建shell脚本文件

- cd /usr/local/bin进入目录
- touch log\_analysis.sh创建脚本文件
- chmod +x log\_analysis.sh 为脚本添加权限

```
[root@localhost ~]# cd /usr/local/bin  
[root@localhost bin]# touch log_analysis.sh  
[root@localhost bin]#
```

命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双引号)

历史

选项

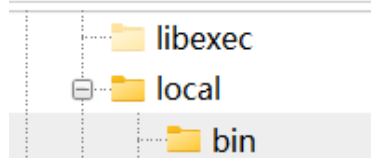


文件

命令

/usr/local/bin

历史



文件名

大小

类型

修改时间

log\_analysis.sh

0 SH 源文件

2024/06/12 16:21

```
[root@localhost bin]# chmod +x log_analysis.sh
```

### 3.2编写shell脚本

```
1 #!/bin/bash  
2
```

```

3 # Nginx日志文件路径
4 LOG_FILE="/var/log/nginx/access.log"
5
6 # MySQL连接信息
7 DB_HOST="192.168.241.132"
8 DB_USER="root"
9 DB_PASS="Mzq0820^"
10 DB_NAME="nginx_logs"
11
12 # 获取当前时间, 用于统计的时间戳
13 CURRENT_TIME=$(date +%s)
14 CURRENT_TIME1=$(date '+%Y-%m-%d %H:%M:%S')
15
16 # 解析日志文件并统计QPS和HTTP状态码200和500+的数量
17 QPS=$(grep "$(date -d '60 seconds ago' '+%d/%b/%Y:%H:%M')" $LOG_FILE | wc -l)
18
19 HTTP_200=$(grep "$(date -d '60 seconds ago' '+%d/%b/%Y:%H:%M')" $LOG_FILE |
    grep " 200 " | wc -l)
20
21 HTTP_500=$(grep "$(date -d '60 seconds ago' '+%d/%b/%Y:%H:%M')" $LOG_FILE |
    grep " 5[0-9][0-9] " | wc -l)
22
23
24 # 创建数据库和表 (如果不存在)
25 mysql -u $DB_USER -p$DB_PASS << EOF
26 CREATE DATABASE IF NOT EXISTS $DB_NAME;
27 USE $DB_NAME;
28 CREATE TABLE IF NOT EXISTS nginx_access_logs (
29     id INT AUTO_INCREMENT PRIMARY KEY,
30     timestamp DATETIME NOT NULL,
31     qps FLOAT NOT NULL,
32     http_200 INT NOT NULL,
33     http_500 INT NOT NULL
34 );
35 EOF
36
37 # 插入统计结果到MySQL数据库
38 mysql -u $DB_USER -p$DB_PASS $DB_NAME << EOF
39 INSERT INTO nginx_access_logs (timestamp, qps, http_200, http_500)
40 VALUES ('$CURRENT_TIME1', '$QPS', '$HTTP_200', '$HTTP_500');
41 EOF

```

## 4.设置定时运行脚本

### 4.1 通过 `crontab -e` 命令打开crontab编辑器

## 4.2编写定时任务

一分钟执行一次shell脚本。

```
*/1 * * * * /usr/local/bin/log_analysis.sh
```

```
[root@localhost ~]# crontab -e  
no crontab for root - using an empty one  
crontab: installing new crontab
```

## 4.3开始执行命令

通过 `systemctl start crond` 命令开始执行定时任务。

```
[root@localhost ~]# systemctl start crond  
您在 /var/spool/mail/root 中有新邮件
```

## 4.4查看数据库

25	2024-06-11 05:22:01	4	2	1
26	2024-06-11 05:23:02	5	1	3
27	2024-06-11 05:24:01	7	4	0