

山东大学计算机科学与技术学院云计算技术课程实验报告

- 学号 : 201900161104 姓名:张浩男 班级:2019级4班
- 实验题目 : 实现在云服务器上的负载均衡
- 实验学时 : 6 实验日期:2022/05/09

实验目的

在云端用docker部署nginx和tomcat实现负载均衡

具体包括:

1. 要求部署一台Nginx和三台Tomcat服务器
2. Nginx需要实现三种策略:
 1. 轮询;
 2. 权重, 三台服务器的权重为1, 3, 5;
 3. IP Hash。

最终实现的效果是, 本地电脑通过网页访问云服务器时, 网页能够体现Nginx三种策略的结果。

实验环境

硬件环境

联网的计算机一台

软件环境

华为云

XShell

Docker

Nginx

Tomcat

实验内容与步骤

配置一台云服务器

这里选择通过华为云配置云服务器。

1. 基础配置

这里选择的内容如下:

- 计费模式 包年包月
- 区域 华北-北京四
- 可用区 可用区1
- CPU架构 x86计算
- 规格

1 | s6.large.4 2vCPUs 8GiB IntelCascade 0.2/1.5Gits/s 15万PPS IPV6 是

- 镜像
公共镜像 : CentOS7.5 64bit(40GB)
- 系统盘 高IO40GiB

完整内容如下：

华为云 控制台

弹性云服务器

1 基础配置 2 网络配置 3 高级配置 4 确认配置

计费模式: **包年包月** 按需计费 竞价计费

区域: 华北-北京四 推荐区域 华北-乌兰察布一 西南-贵阳一 华北-北京四 华南-广州 华东-上海一

可用区: 随机分配 **可用区1** 可用区2 可用区3 可用区7

CPU架构: **x86计算** 鲲鹏计算

规格: 全部系列 vCPUs 2vCPUs 内存 8GiB 规格名称

规格名称	vCPUs	内存	CPU	基准 / 最大带宽	内网收发包	特性	IPv6	规格参考价
sn3.large.4 (已售罄)	2vCPUs	8GiB	Intel SkyLake ...	0.35 / 1.5 Gbit/s	15万PPS	--	是	¥226.40/月
s6.large.4	2vCPUs	8GiB	Intel Cascade ...	0.2 / 1.5 Gbit/s	15万PPS	--	是	¥215.60/月
ir3.large.4	2vCPUs	8GiB	Intel Cascade ...	1.2 / 4 Gbit/s	40万PPS	--	否	¥307.37/月

购买量: 1 台 1个月 配置费用: **¥283.60**

下一步: 网络配置

弹性公网IP: ☒ 现在购买 ☐ 使用已有 ☐ 暂不购买

线路: **全动态BGP** 静态BGP

公网带宽: **按带宽计费** 按流量计费 加入共享带宽

带宽大小: 1 2 5 10 100 200 自定义 2 带宽范围: 1-2,000 Mbit/s

购买量: 1 台 1个月 配置费用: **¥283.60**

上一步 下一步: 高级配置

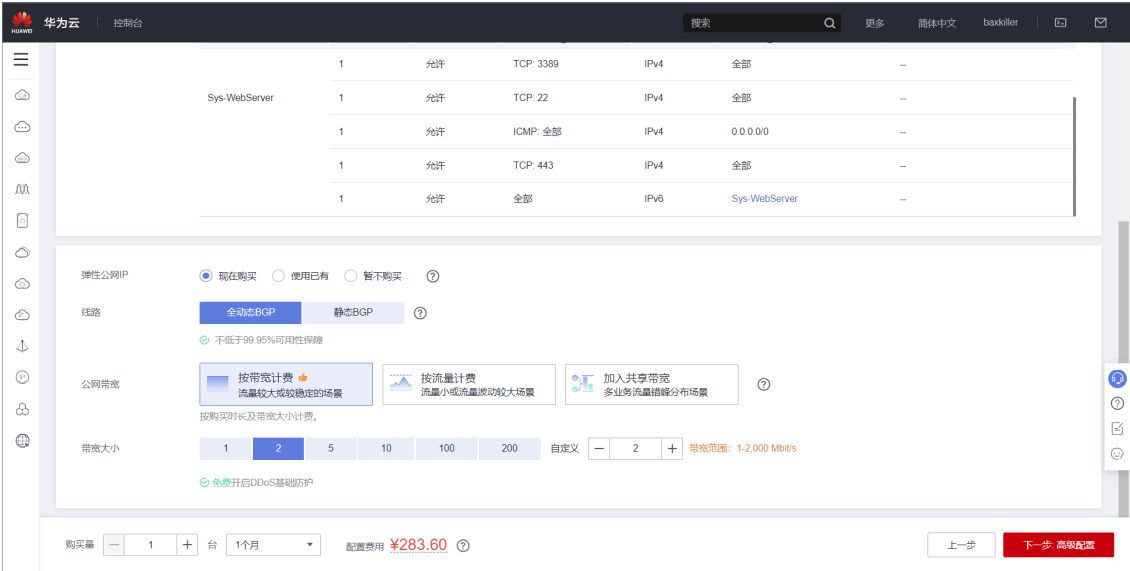
2. 网络配置

虚拟私有云选择 **subnet-default**

弹性公网IP可以选择现在购买.全动态BGP

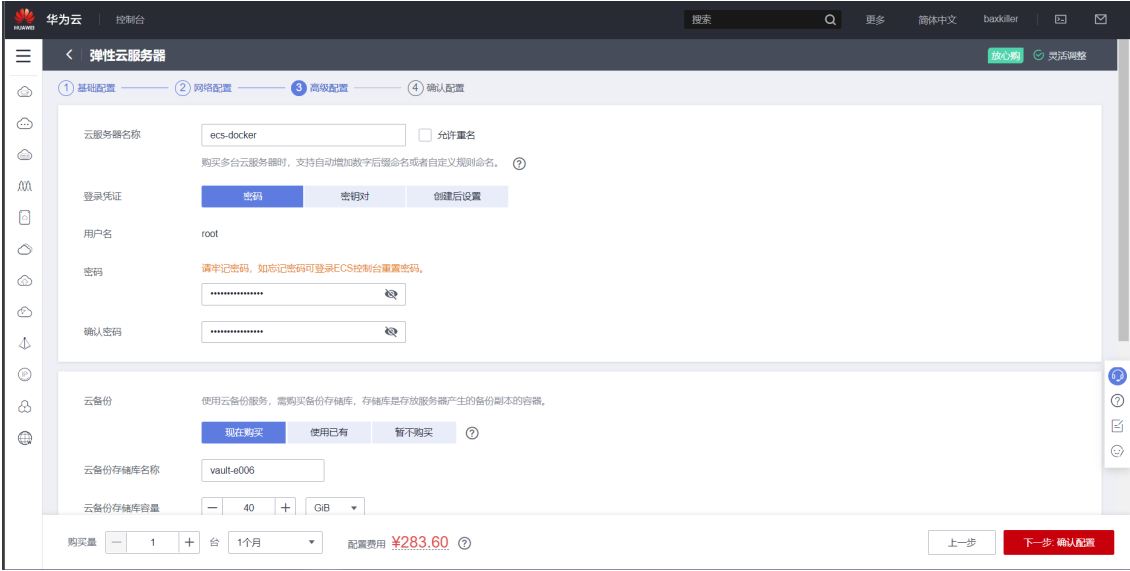
公网带宽选择按照带宽计费

带宽大小2,如下图所示:

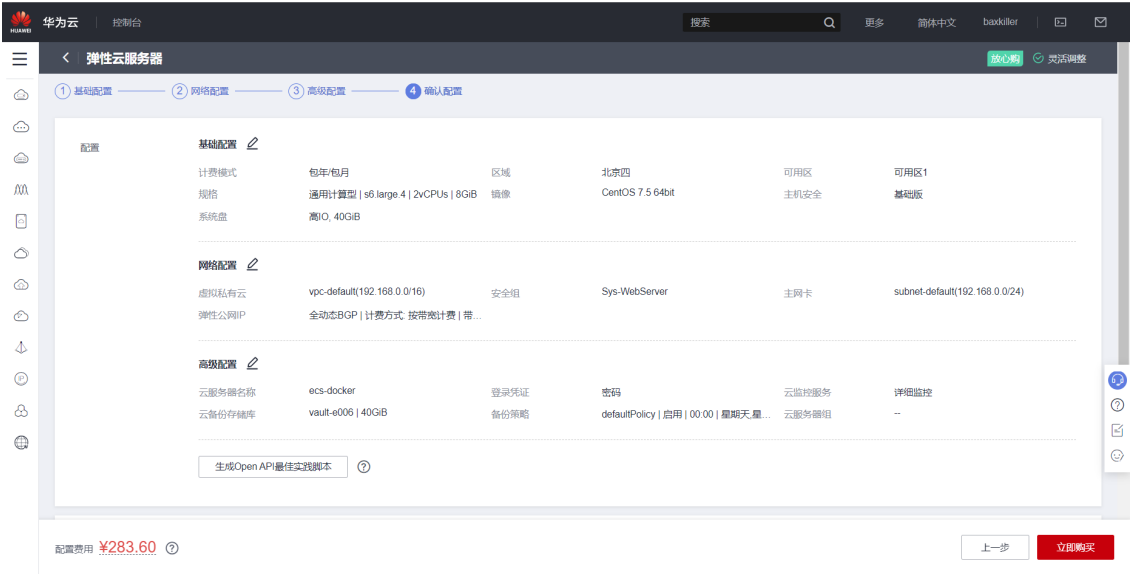


3. 高级配置

在此处设置云服务器名称,登陆凭证为密码

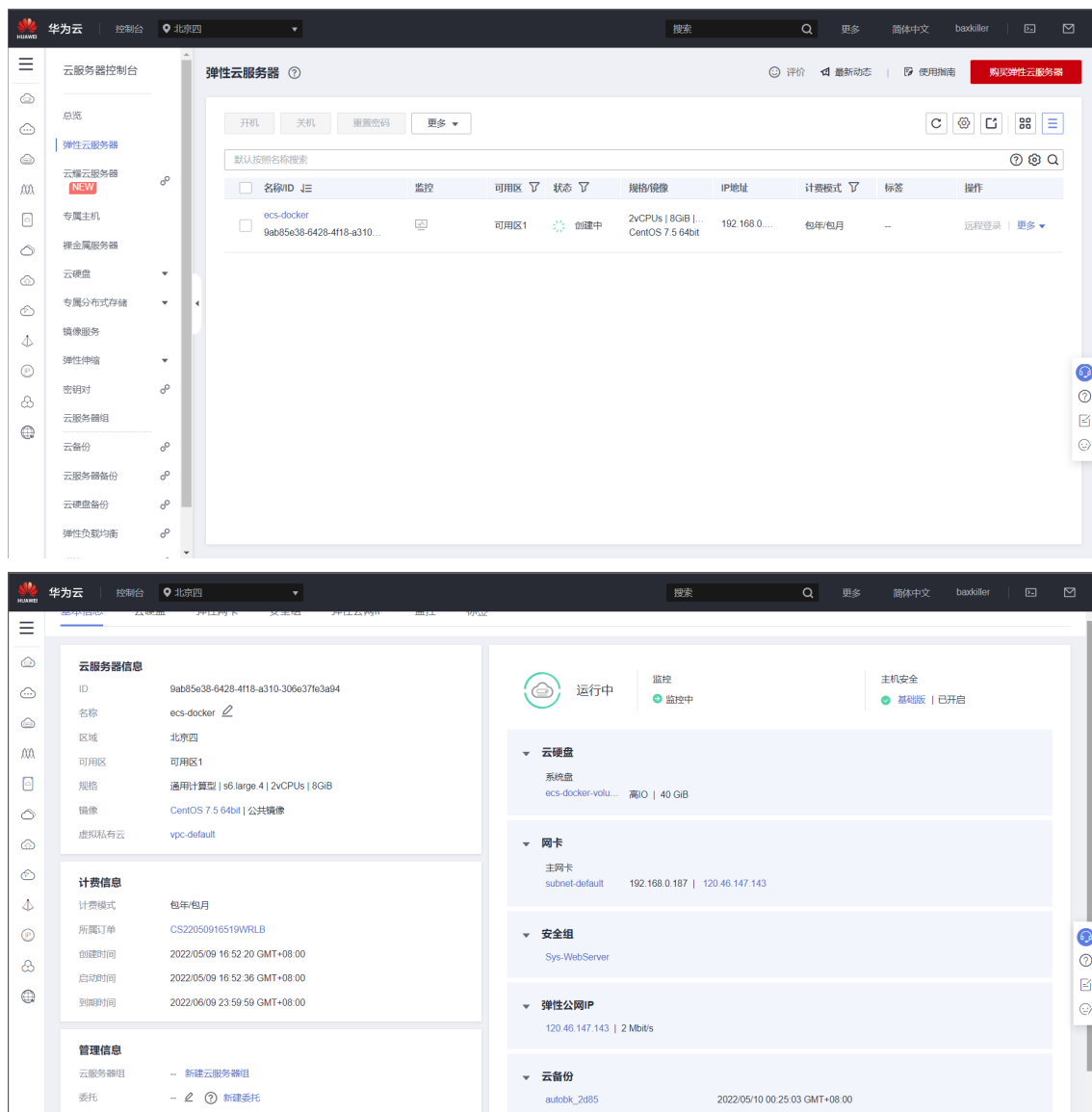


4. 确认配置 此处可以看到先前配置的所有内容,如下所示:



5. 购买

购买完成后的界面应当如下所示:



服务器的链接与环境配置

服务器的链接这里选择使用XShell进行连接.

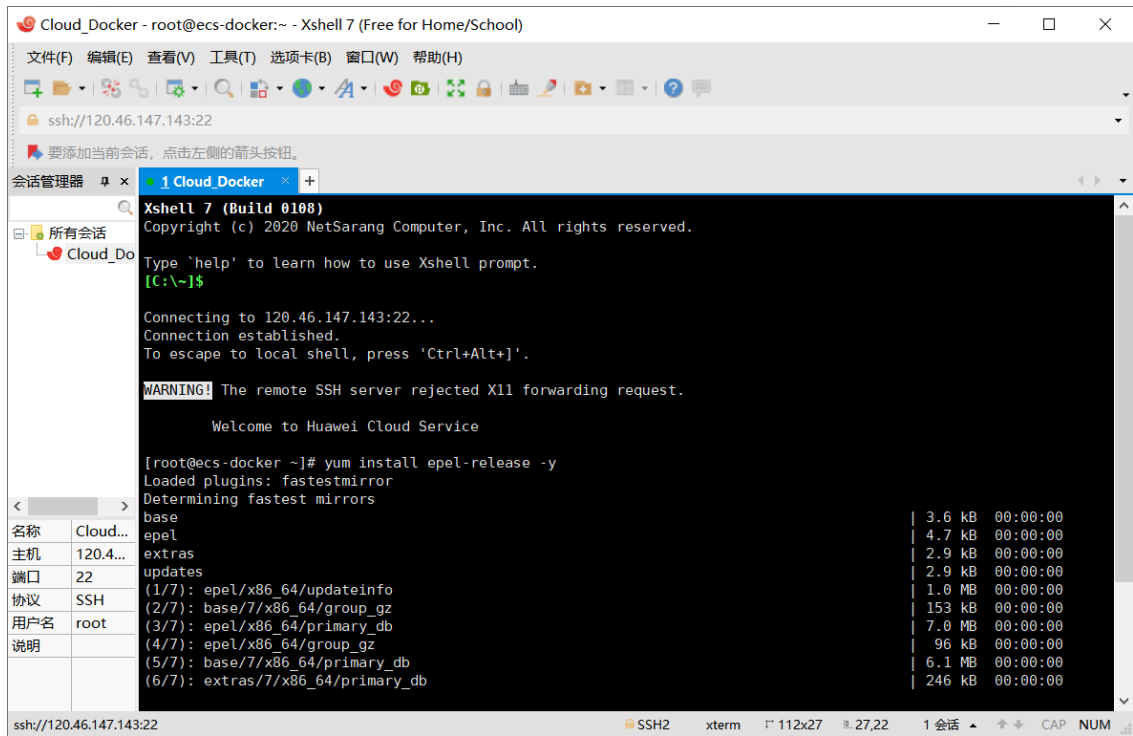
链接

通过新建,输入服务器的公网IP,然后输入用户名和密码,即可成功链接.

安装Docker

1. 添加yml源

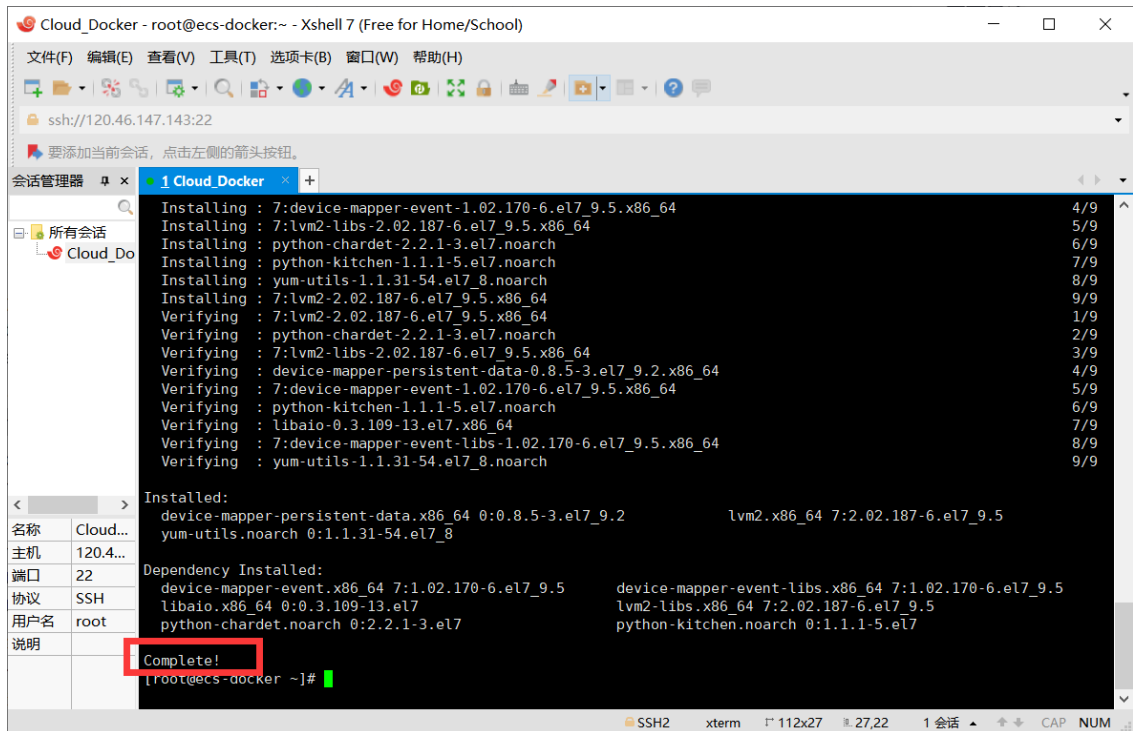
```
1 yum install epel-release -y
2 yum clean all
```



2. 安装yum-utils

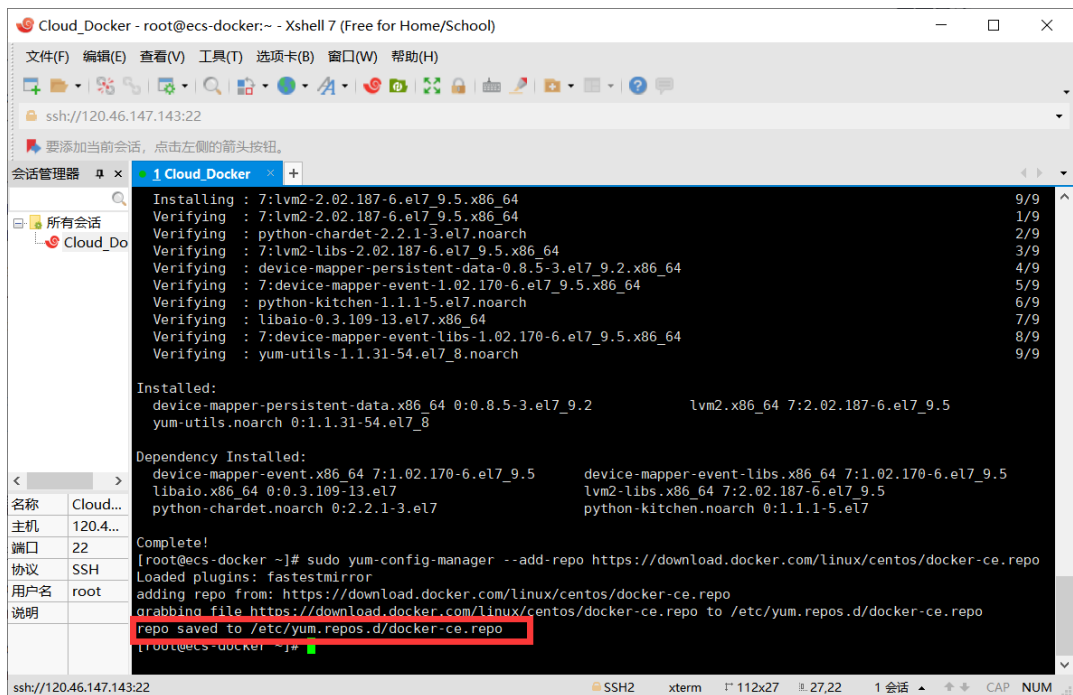
```
1 | sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

安装成功如下:



3. 设置docker yum源

```
1 | sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
```

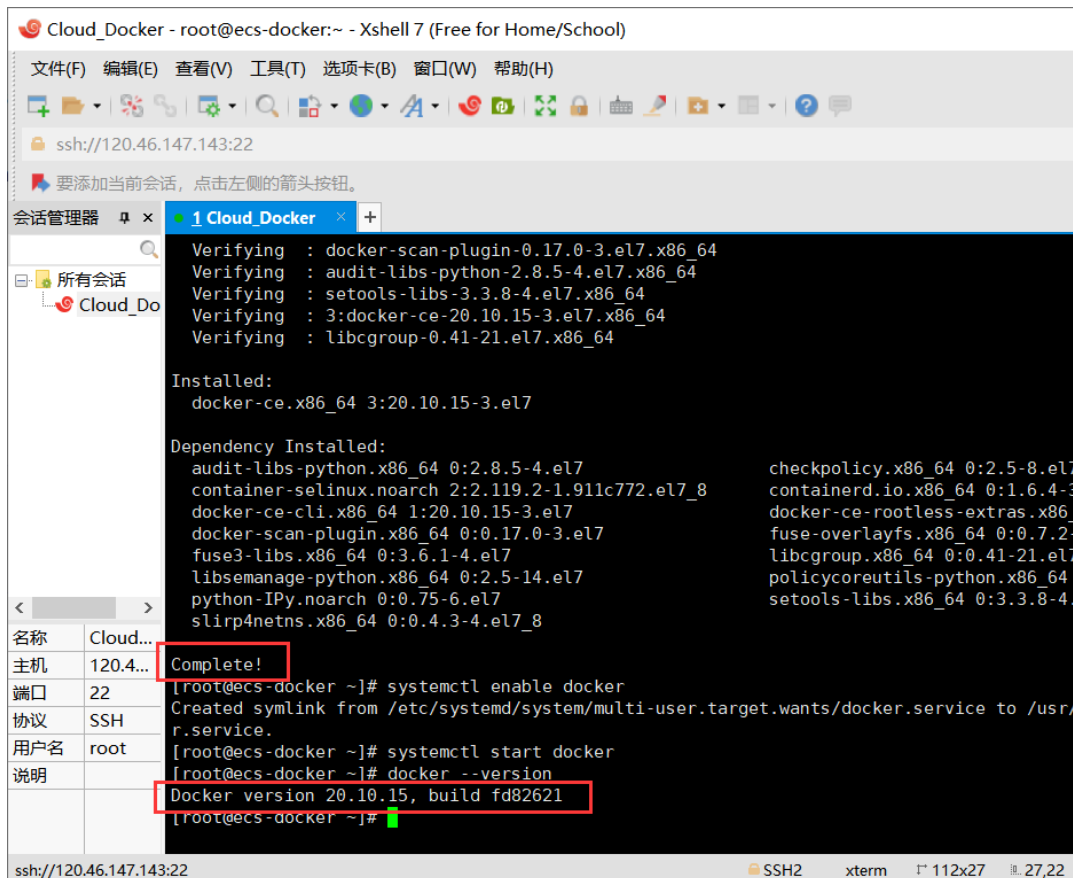


4. 安装并运行Docker。

```
1 sudo yum install docker-ce
2 systemctl enable docker
3 systemctl start docker
```

检查安装结果

```
1 docker --version
```

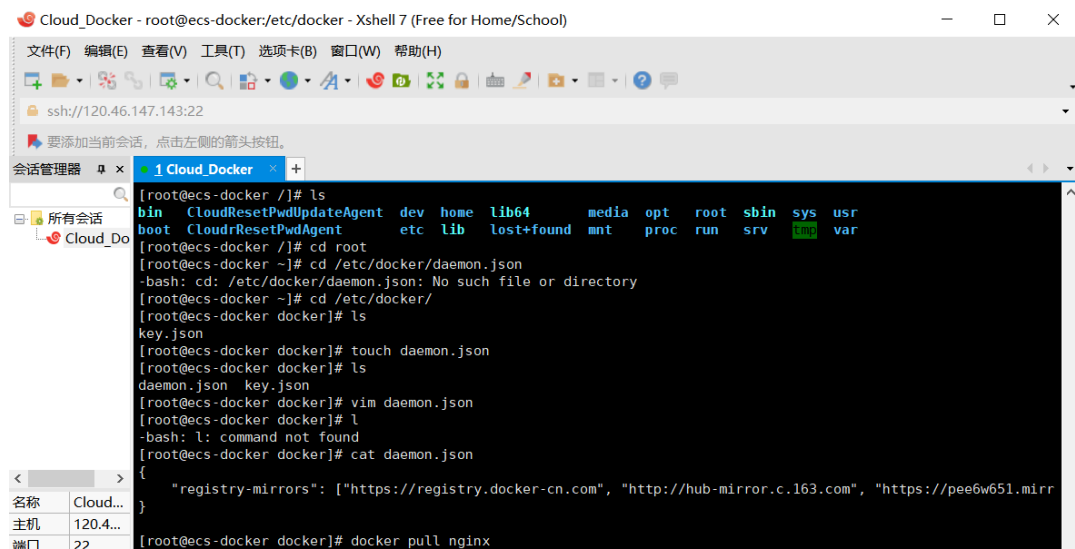


5. 配置docker镜像仓库(编辑对应json文件)

路径: `/etc/docker/daemon.json`

内容:

```
1 {
2     "registry-mirrors":
3     ["https://registry.docker-cn.com",
4      "http://hub-mirror.c.163.com",
5      "https://pee6w651.mirror.aliyuncs.com"]
6 }
```

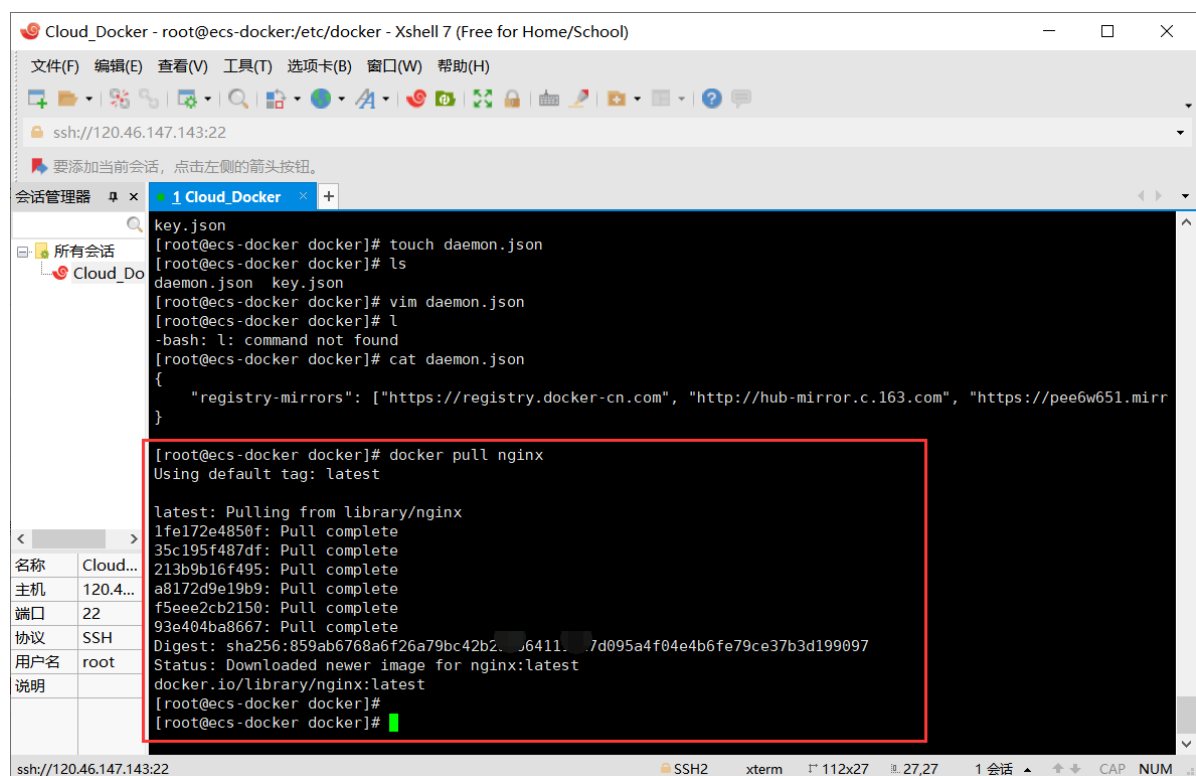


```
Cloud_Docker - root@ecs-docker:/etc/docker - Xshell 7 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://120.46.147.143:22
要添加当前会话，点击左侧的箭头按钮。
会话管理器
1 Cloud_Docker
所有会话
Cloud_Docker
[root@ecs-docker /]# ls
bin CloudResetPwdUpdateAgent dev home lib64 media opt root sbin sys usr
boot CloudResetPwdUpdateAgent etc lib lost+found mnt proc run srv tmp var
[root@ecs-docker /]# cd /etc/docker/
[root@ecs-docker /etc/docker]# cd /etc/docker/
[root@ecs-docker /etc/docker]# ls
key.json
[root@ecs-docker /etc/docker]# touch daemon.json
[root@ecs-docker /etc/docker]# ls
daemon.json key.json
[root@ecs-docker /etc/docker]# vim daemon.json
[root@ecs-docker /etc/docker]# l
-bash: l: command not found
[root@ecs-docker /etc/docker]# cat daemon.json
{
  "registry-mirrors": ["https://registry.docker-cn.com", "http://hub-mirror.c.163.com", "https://pee6w651.mirr
}
[root@ecs-docker /etc/docker]# docker pull nginx
```

配置nginx与tomcat服务器

1. 下载nginx和tomcat的镜像

```
1 docker pull nginx
2 docker pull tomcat
```



```
Cloud_Docker - root@ecs-docker:/etc/docker - Xshell 7 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://120.46.147.143:22
要添加当前会话，点击左侧的箭头按钮。
会话管理器
1 Cloud_Docker
所有会话
Cloud_Docker
[root@ecs-docker /etc/docker]# touch daemon.json
[root@ecs-docker /etc/docker]# ls
daemon.json key.json
[root@ecs-docker /etc/docker]# vim daemon.json
[root@ecs-docker /etc/docker]# l
-bash: l: command not found
[root@ecs-docker /etc/docker]# cat daemon.json
{
  "registry-mirrors": ["https://registry.docker-cn.com", "http://hub-mirror.c.163.com", "https://pee6w651.mirr
}
[root@ecs-docker /etc/docker]# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
1fe172e4850f: Pull complete
35c195f487df: Pull complete
213b9b16f495: Pull complete
a8172d9e19b9: Pull complete
f5eee2cb2150: Pull complete
93e404ba8667: Pull complete
Digest: sha256:859ab6768a6f26a79bc42b2.06411.7d095a4f04e4b6fe79ce37b3d199097
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[root@ecs-docker /etc/docker]#
[root@ecs-docker /etc/docker]#
```

The screenshot shows a terminal window titled "Cloud_Docker - root@ecs-docker:/etc/docker - Xshell 7 (Free for Home/School)". The terminal output shows the following commands and results:

```
1fe172e4850f: Pull complete
35c195f487df: Pull complete
213b9b16f495: Pull complete
a8172d9e19b9: Pull complete
f5eee2cb2150: Pull complete
93e404ba8667: Pull complete
Digest: sha256:859ab6768a6f26a79bc42b231664111317d095a4f04e4b6fe79ce37b3d199097
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[root@ecs-docker docker]#
[root@ecs-docker docker]# docker pull tomcat
Using default tag: latest
latest: Pulling from library/tomcat
6aefca2dc61d: Pull complete
967757d56527: Pull complete
c357e2c68cb3: Pull complete
c766e27afb21: Pull complete
a747e81e6111: Pull complete
2859d18181fd: Pull complete
9706c6a496b8: Pull complete
1fcc3b6a96c1: Pull complete
21609621b8c4: Pull complete
d000f8b4b3ce: Pull complete
Digest: sha256:3a407bd33b2e42f8e7ba8f736765cda37cea1cdf40a97262169f18e3edcb9acb
Status: Downloaded newer image for tomcat:latest
docker.io/library/tomcat:latest
[root@ecs-docker docker]#
```

The terminal window also shows a sidebar with a list of sessions and a table of session details.

名称	Cloud...
主机	120.4...
端口	22
协议	SSH
用户名	root
说明	

2. 创建nginx和tomcat本地目录 稍后挂载到docker容器上

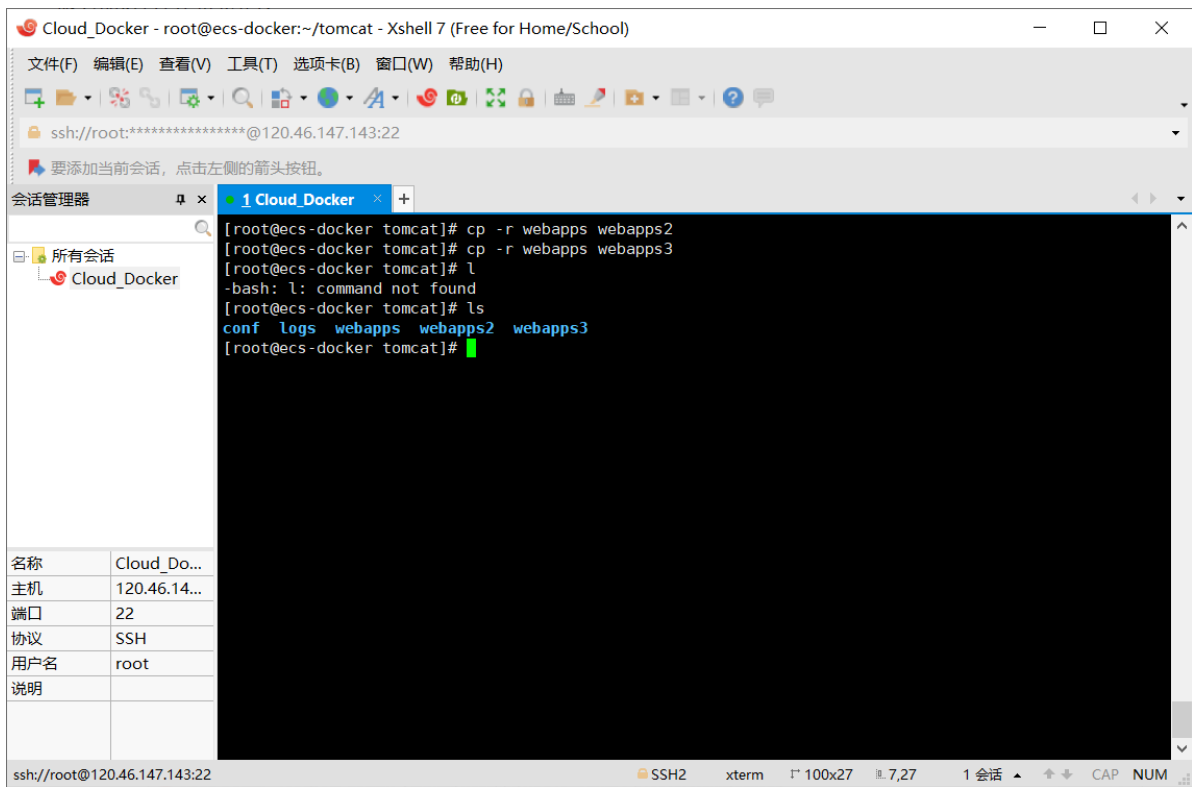
这里为了对三个tomcat对应的服务器主页面进行区分,在原有 **webapps** 文件的基础上,再创建两个 **webapps** 文件,用于将不同的tomcat服务器进行各自的对应.

The screenshot shows a terminal window titled "Cloud_Docker - root@ecs-docker:~/tomcat/webapps/ROOT - Xshell 7 (Free for Home/School)". The terminal output shows the following commands and results:

```
[root@ecs-docker ~]# cd Dockerfile/
[root@ecs-docker Dockerfile]# touch Dockerfile
[root@ecs-docker Dockerfile]# vim Dockerfile
[root@ecs-docker Dockerfile]# systemctl start docker.service
[root@ecs-docker Dockerfile]# docker pull centos
Using default tag: latest
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest
[root@ecs-docker Dockerfile]# cd
[root@ecs-docker ~]# mkdir -p ~/nginx/www ~/nginx/conf/ ~/nginx/logs
[root@ecs-docker ~]# mkdir -p ~/tomcat/webapps/ROOT ~/tomcat/conf ~/tomcat/logs
[root@ecs-docker ~]# ls
Dockerfile nginx tomcat
[root@ecs-docker ~]# cd tomcat/webapps/ROOT/
[root@ecs-docker ROOT]# ls
[root@ecs-docker ROOT]# touch index.html
[root@ecs-docker ROOT]# ls
index.html
[root@ecs-docker ROOT]# vim index.html
[root@ecs-docker ROOT]# docker run -d --name tomcat1 -v ~/tomcat/webapps:/usr/local/tomcat/webapps tomcat
94d85a2311bbd85827f9b354f1f57ab9e6c612a5ec4af21edf8493cb98e7552c
[root@ecs-docker ROOT]# docker run -d --name tomcat2 -v ~/tomcat/webapps:/usr/local/tomcat/webapps tomcat
f663b6179a9c1a73646614c7a09cbf320c6c00f1f3378f99eb7fda68796171d9
[root@ecs-docker ROOT]#
```

The terminal window also shows a sidebar with a list of sessions and a table of session details.

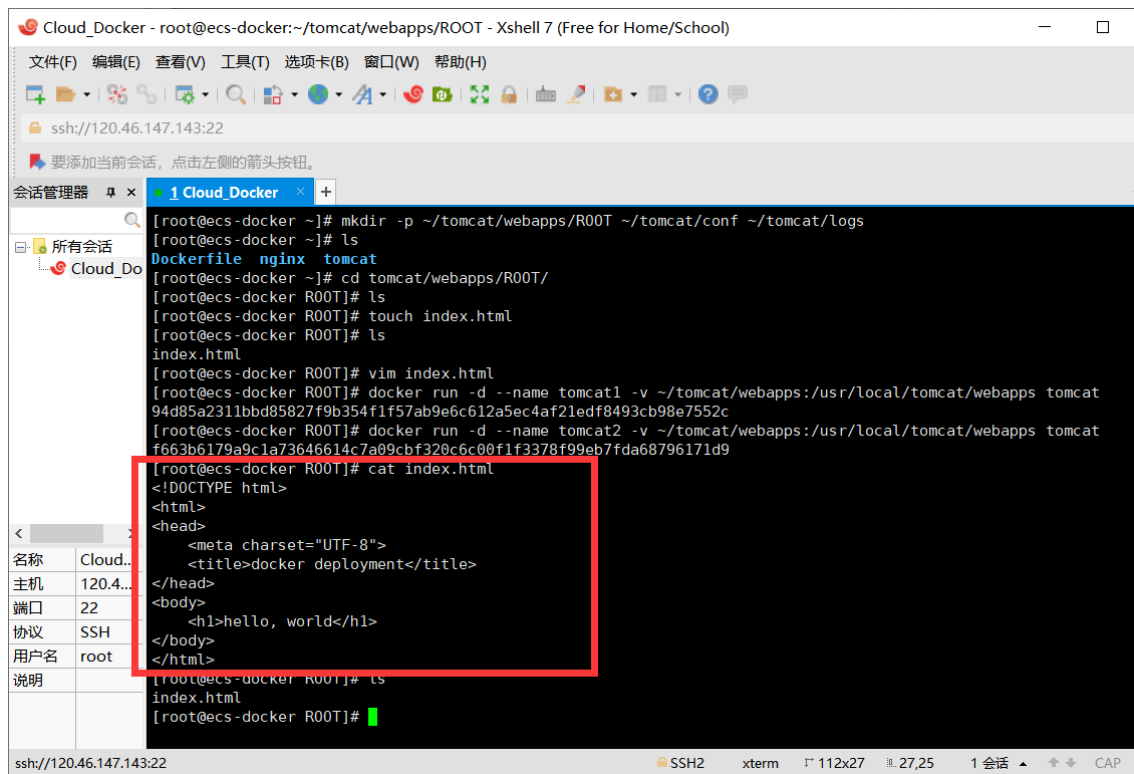
名称	Cloud...
主机	120.4...
端口	22
协议	SSH
用户名	root
说明	



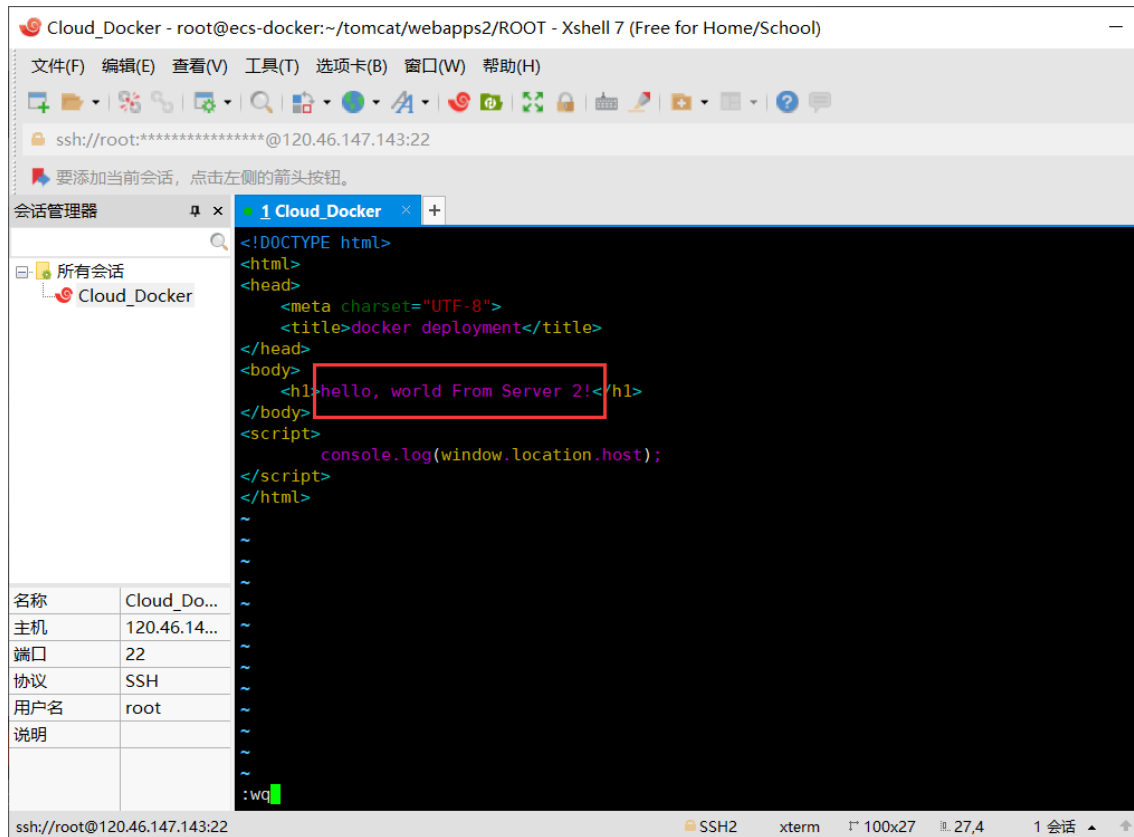
3. 编写 `index.html`

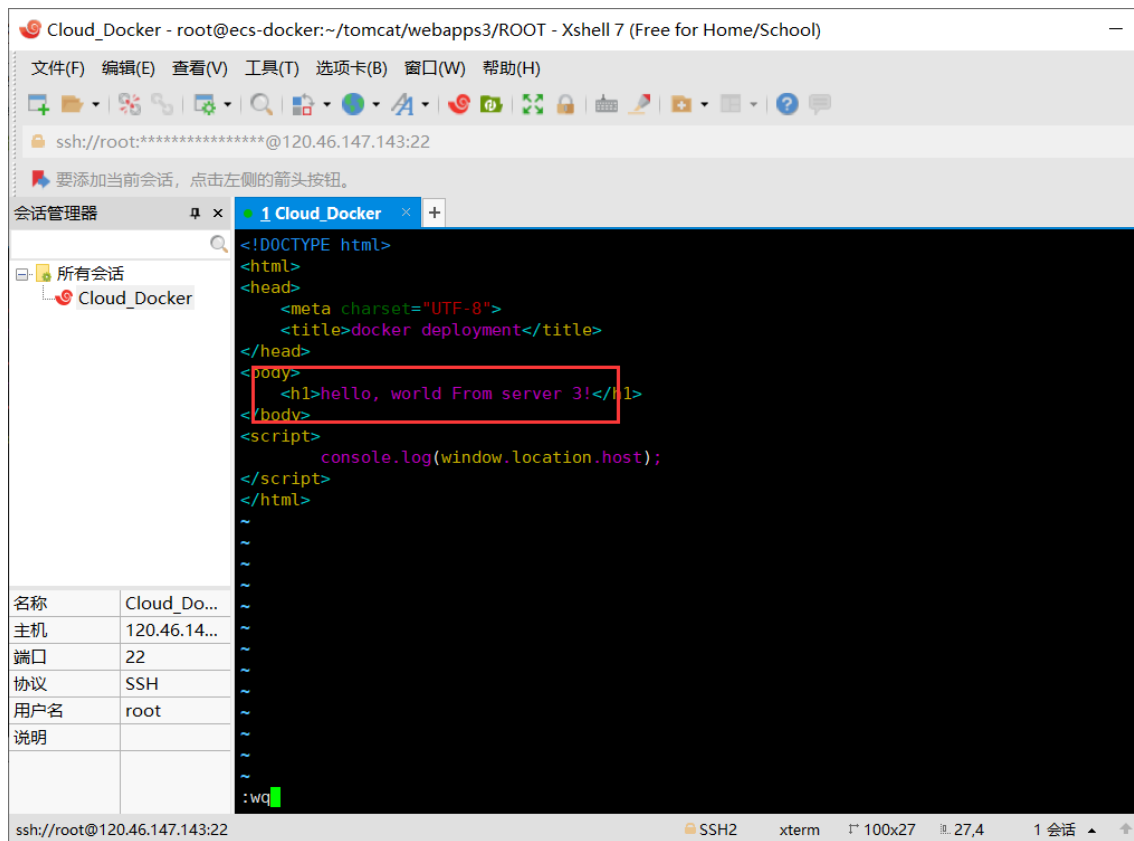
内容如下:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>docker deployment</title>
6 </head>
7 <body>
8     <h1>hello, world</h1>
9 </body>
10 </html>
```



将该文件分别复制到另外两个 **webapps** 文件夹,并进行一定的修改以显现差异:



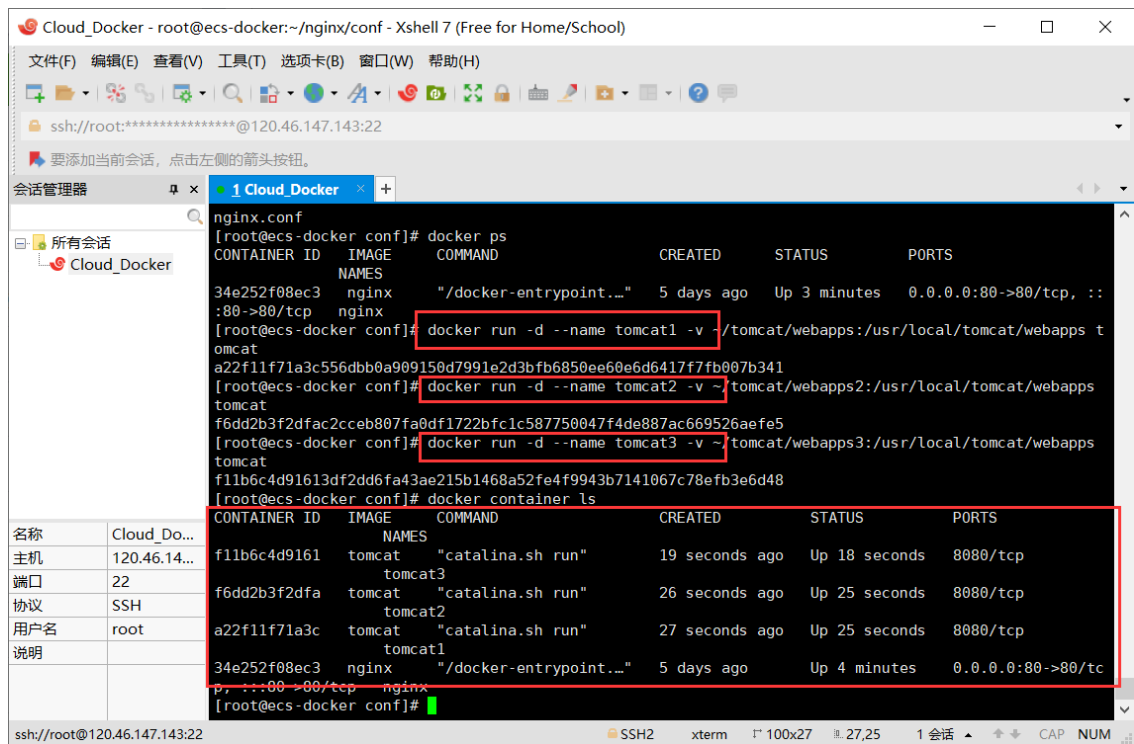


4. 启动三个tomcat容器

注意这里的三个tomcat挂载的目录需要是不同的!

- 1 `docker run -d --name tomcat1 -v`
`~/tomcat/webapps:/usr/local/tomcat/webapps tomcat`
- 2 `docker run -d --name tomcat2 -v`
`~/tomcat/webapps2:/usr/local/tomcat/webapps tomcat`
- 3 `docker run -d --name tomcat3 -v`
`~/tomcat/webapps3:/usr/local/tomcat/webapps tomcat`

启动完成后如下所示:

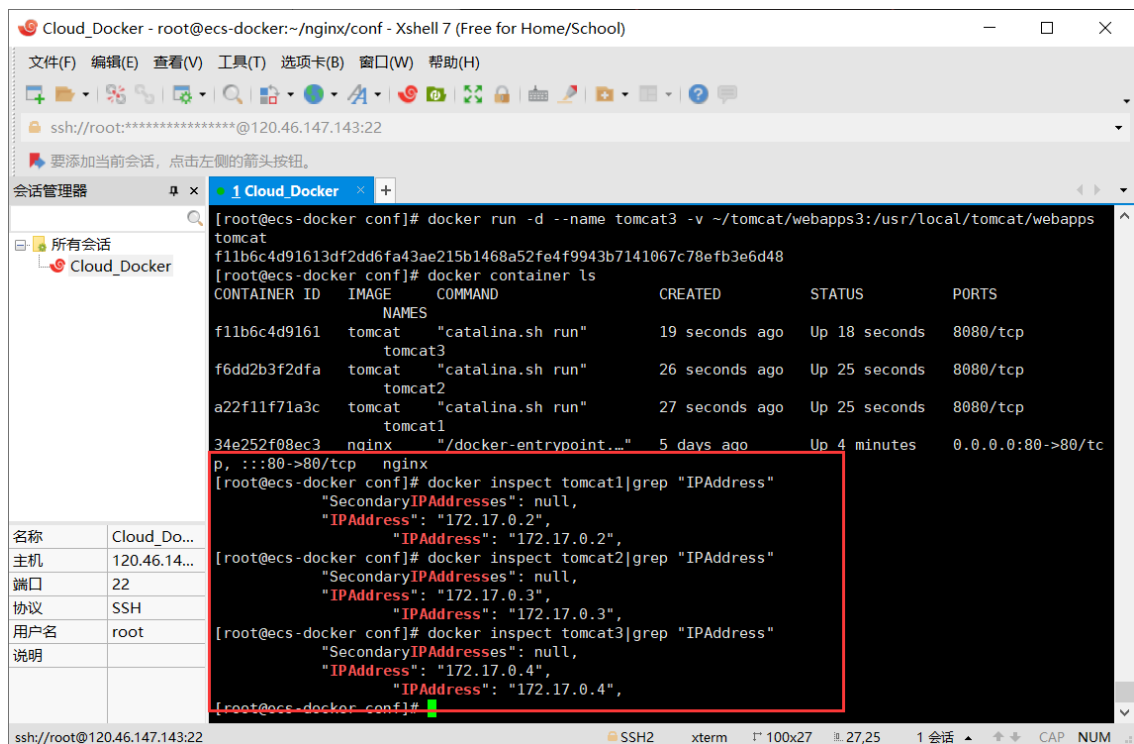


5. 获取tomcat容器IP，获取到的IP将配置到Nginx的配置文件中：

IP获取：

```

1 docker inspect tomcat1|grep "IPAddress"
2 docker inspect tomcat2|grep "IPAddress"
3 docker inspect tomcat3|grep "IPAddress"
  
```



6. 配置相关文件

路径: nginx/conf/nginx.conf

nginx配置文件:

```

1 user nginx;
2 worker_processes 1;
  
```

```

3
4 error_log /var/log/nginx/error.log warn;
5 pid /var/run/nginx.pid;
6
7
8 events {
9     worker_connections 1024;
10 }
11
12
13 http {
14     include /etc/nginx/mime.types;
15     default_type application/octet-stream;
16
17     log_format main '$remote_addr - $remote_user [$time_local]
"$request" '
18                     '$status $body_bytes_sent "$http_referer" '
19                     '"$http_user_agent" "$http_x_forwarded_for"
"$upstream_addr"';
20
21     access_log /var/log/nginx/access.log main;
22
23     sendfile on;
24     #tcp_nopush on;
25
26     keepalive_timeout 65;
27
28     #gzip on;
29
30     upstream tomcat {
31         server 172.17.0.2:8080;
32         server 172.17.0.3:8080;
33         server 172.17.0.4:8080;
34     }
35
36     server {
37         listen 80;
38         server_name localhost;
39
40         location / {
41             proxy_pass http://tomcat;
42             proxy_redirect off;
43             index index.html index.htm;
44             proxy_set_header Host $host;
45             proxy_set_header X-Real-IP $remote_addr;
46             proxy_set_header X-Real-Port $remote_port;
47             proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
48         }
49
50         location /static/ {
51             alias /usr/share/nginx/html/;
52         }
53     }
54
55     include /etc/nginx/conf.d/*.conf;
56 }

```

7. 启动nginx:

```
1 docker run -d -p 80:80 --name nginx -v ~/nginx/www:/usr/share/nginx/html -v ~/nginx/conf/nginx.conf:/etc/nginx/nginx.conf -v ~/nginx/logs:/var/log/nginx nginx
```

轮询

• 说明

由于nginx的配置文件中,如果没添加额外的内容,默认的策略就是轮询,因此这里不再展示对应配置文件的内容(与上面提到的配置文件是相同的)

- **测试**

尝试多次访问服务器地址:[docker deployment](#)

查看nginx日志: `access.log` 文件

根据下图,不难看出,在多次访问网站时,网站反馈得到的具体访问处理服务器是平均分配且顺序轮询的.

即第一次访问对应于设置中给定的第一个服务器地址 172.17.0.2:8000,第二次访问对应于设置中给定的第二个服务器地址 172.17.0.3:8000,第三次依次类推...

```

1 Cloud_Docker x +
223.80.110.108 - - [10/May/2022:16:21:15 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
/101.0.1210.39" "-" "172.17.0.2:8080"
223.80.110.108 - - [10/May/2022:16:21:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
/101.0.1210.39" "-" "172.17.0.3:8080"
223.80.110.108 - - [10/May/2022:16:21:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
/101.0.1210.39" "-" "172.17.0.4:8080"
223.80.110.108 - - [10/May/2022:16:21:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
/101.0.1210.39" "-" "172.17.0.2:8080"
223.80.110.108 - - [10/May/2022:16:21:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
223.80.110.108 - - [10/May/2022:16:21:15 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.2:8080"
223.80.110.108 - - [10/May/2022:16:21:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.3:8080"
223.80.110.108 - - [10/May/2022:16:21:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.4:8080"
223.80.110.108 - - [10/May/2022:16:21:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.2:8080"
223.80.110.108 - - [10/May/2022:16:21:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.3:8080"
223.80.110.108 - - [10/May/2022:16:21:17 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.4:8080"
223.80.110.108 - - [10/May/2022:16:21:17 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.2:8080"
223.80.110.108 - - [10/May/2022:16:21:17 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.3:8080"
223.80.110.108 - - [10/May/2022:16:21:17 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.4:8080"
223.80.110.108 - - [10/May/2022:16:21:18 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36 Edg/101.0.1210.39" "-" "
172.17.0.2:8080"
109.237.103.9 - - [10/May/2022:17:10:43 +0000] "\x16\x03\x01\x01D\x01\x00\x01@\x03\x03ze\xFB\xD0\xC0 z\x09\x

```

反应到网页上,即有以下的结果:

以下三个页面依次轮流出现:





权重

- 修改

按照实验指导书说明,设定三台服务器对应的权重信息为 **1,2,3**

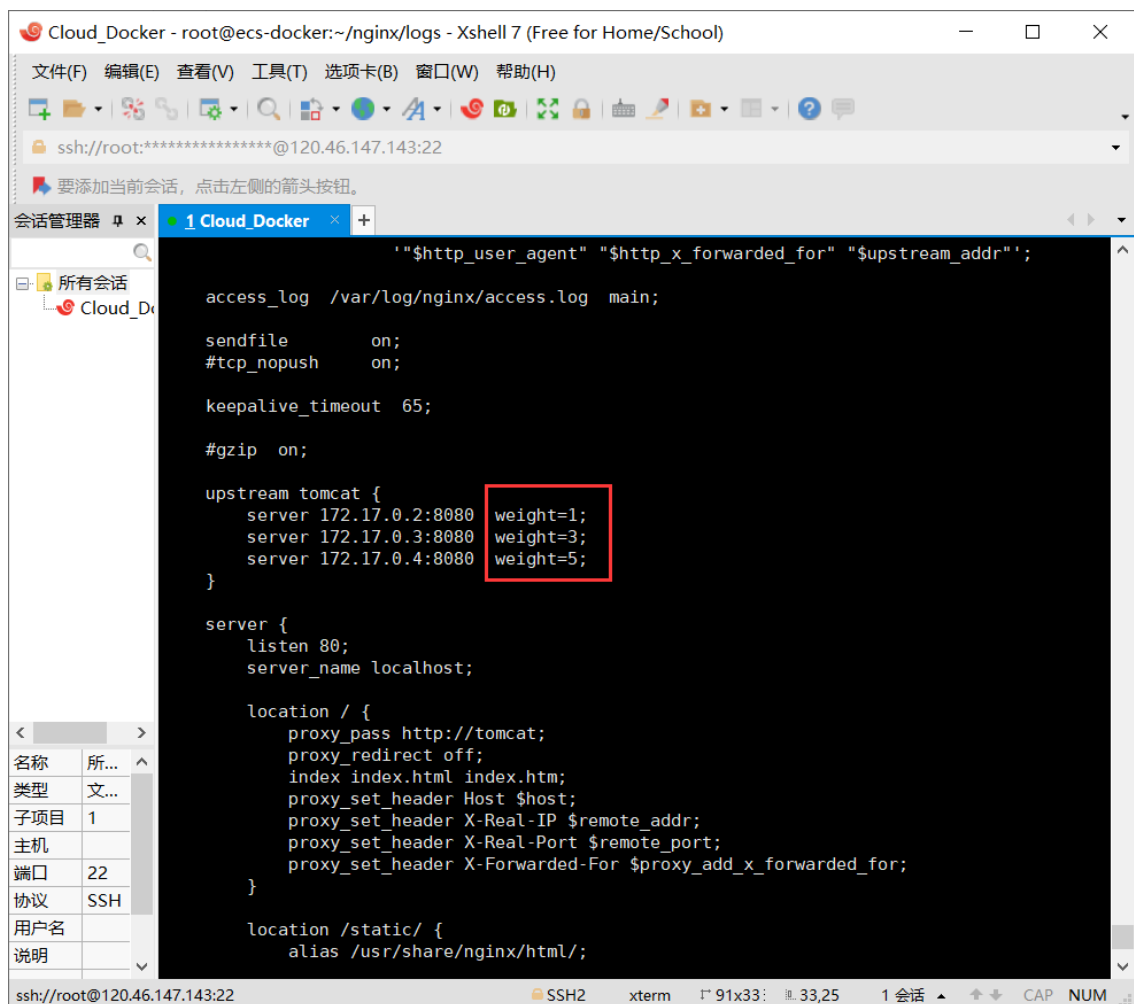
修改完成后的 `~/nginx/conf/nginx.json` 内部代码如下所示:

```
1 user nginx;
2 worker_processes 1;
3
4 error_log /var/log/nginx/error.log warn;
5 pid /var/run/nginx.pid;
6
7
8 events {
9     worker_connections 1024;
10 }
11
12 http {
13     include /etc/nginx/mime.types;
14     default_type application/octet-stream;
15
16     log_format main '$remote_addr - $remote_user [$time_local]
"$request" '
17                     '$status $body_bytes_sent "$http_referer" '
18                     '"$http_user_agent" "$http_x_forwarded_for"
"$upstream_addr"';
19
```



```
20     access_log    /var/log/nginx/access.log  main;
21
22     sendfile      on;
23     #tcp_nopush   on;
24
25     keepalive_timeout 65;
26
27     #gzip on;
28
29     upstream tomcat {
30         server 172.17.0.2:8080 weight=1;
31         server 172.17.0.3:8080 weight=3;
32         server 172.17.0.4:8080 weight=5;
33     }
34
35     server {
36         listen 80;
37         server_name localhost;
38
39         location / {
40             proxy_pass http://tomcat;
41             proxy_redirect off;
42             index index.html index.htm;
43             proxy_set_header Host $host;
44             proxy_set_header X-Real-IP $remote_addr;
45             proxy_set_header X-Real-Port $remote_port;
46             proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
47         }
48
49         location /static/ {
50             alias /usr/share/nginx/html/;
51         }
52     }
53
54     include /etc/nginx/conf.d/*.conf;
55 }
```

与前面的默认相比,新增了如下内容:



```
Cloud_Docker - root@ecs-docker:~/nginx/logs - Xshell 7 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root:*****@120.46.147.143:22
会话管理器
1 Cloud_Docker
名称 所...
类型 文...
子项目 1
主机
端口 22
协议 SSH
用户名
说明
"$http_user_agent" "$http_x_forwarded_for" "$upstream_addr";
access_log /var/log/nginx/access.log main;
sendfile on;
#tcp_nopush on;
keepalive_timeout 65;
#gzip on;
upstream tomcat {
    server 172.17.0.2:8080 weight=1;
    server 172.17.0.3:8080 weight=3;
    server 172.17.0.4:8080 weight=5;
}
server {
    listen 80;
    server_name localhost;
    location / {
        proxy_pass http://tomcat;
        proxy_redirect off;
        index index.html index.htm;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Real-Port $remote_port;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
    location /static/ {
        alias /usr/share/nginx/html/;
    }
}
```

- 重启nginx

输入以下命令重启nginx

```
1 docker restart nginx
```

同时,为了便于观察更改后的请求分配情况,这里将原本请求日志清空:

```
1 rm access.log
2 touch access.log
```

- 测试

在本地物理机多次访问服务器(35次左右)

访问过程中,页面的具体变化表现为:

三个页面随机出现,但是就概率而言,每个页面出现的概率是不同的。权重越大的服务器对应页面出现的概率越高。

查看对应的日志文件 `~/nginx/logs/access.log`,如下图所示.



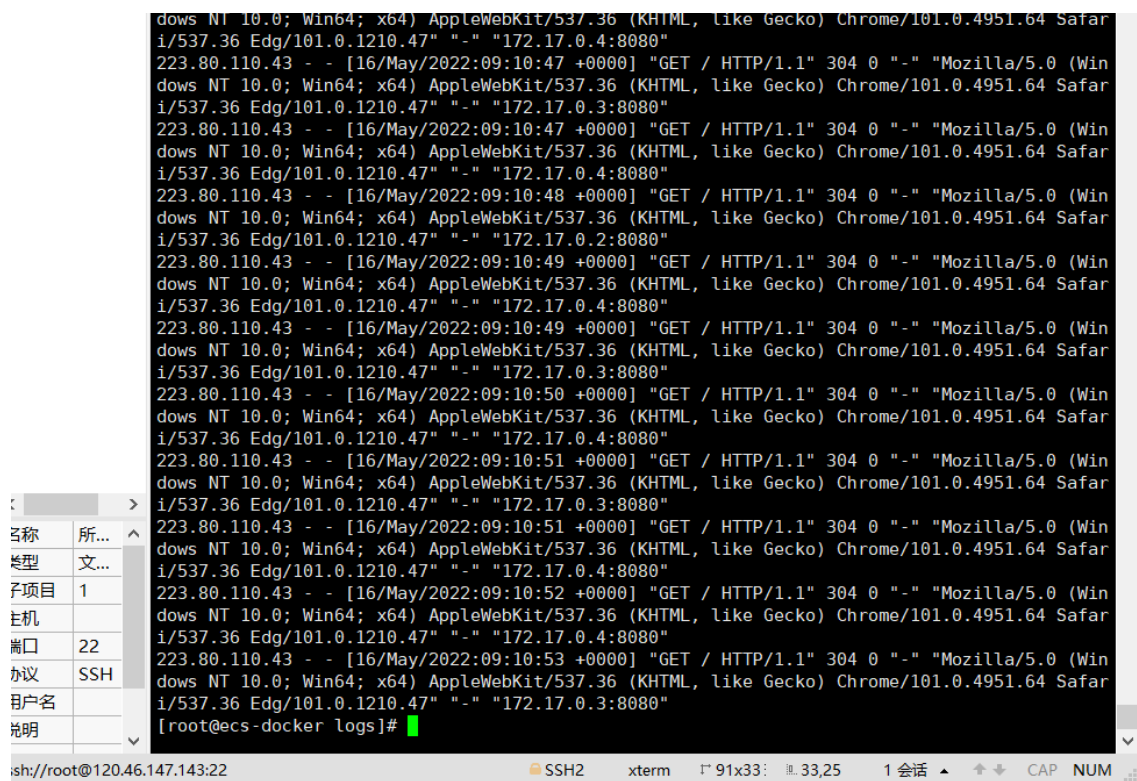
ssh://root:*****@120.46.147.143:22

要添加当前会话, 点击左侧的箭头按钮。

会话管理器 1 Cloud_Docker

所有会话
Cloud_D

```
cat: nginx.conf: No such file or directory
[root@ecs-docker logs]# cat access.log
223.80.110.43 - - [16/May/2022:09:02:37 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:02:38 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:02:39 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:02:40 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.2:8080"
223.80.110.43 - - [16/May/2022:09:02:41 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:02:42 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:02:43 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:32 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:33 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:34 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:10:35 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:36 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.2:8080"
223.80.110.43 - - [16/May/2022:09:10:36 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:37 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:10:38 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:38 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:10:39 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:40 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:10:40 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:41 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.2:8080"
223.80.110.43 - - [16/May/2022:09:10:42 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:43 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:10:43 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:44 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:10:45 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:46 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
```



根据上图中表现出来的日志内容,可以统计得到每个服务器对应的访问次数如下所示:

IP地址	访问次数
172.17.0.4	21
172.17.0.3	12
172.17.0.2	4

可以确定:

- 权重越高分配到需要处理的请求越多。
- 处理的请求个数与对应的权重值成正比

即:

$$4 : 12 : 21 \approx 1 : 3 : 5$$

IP Hash

• 配置修改

这个修改相对容易,只需要在对应配置项中添加一行代码即可:

```
1 ip_hash;
```

修改后的配置文件基本与轮询的配置文件相同,不同内容如下:

```
Cloud_Docker - root@ecs-docker:~/nginx/conf - Xshell 7 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root:*****@120.46.147.143:22
要添加当前会话, 点击左侧的箭头按钮。
会话管理器
1 Cloud_Docker
所有会话
Cloud_Docker
名称 所...
类型 文...
子项目 1
主机
端口 22
协议 SSH
用户名
说明
error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for" "$upstream_addr"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    #tcp_nopush on;

    keepalive_timeout 65;

    #gzip on;

    upstream tomcat {
        ip_hash;
        server 172.17.0.2:8080 ;
        server 172.17.0.3:8080 ;
        server 172.17.0.4:8080 ;
    }
}

"nginx.conf" 57L, 1305C 35,5 12%
ssh://root@120.46.147.143:22 SSH2 xterm 11 91x33 32,5 1 会话 CAP NUM
```

- 重启nginx

该部分内容与上面相同,不过多赘述.

```
Cloud_Docker - root@ecs-docker:~/nginx/conf - Xshell 7 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root:*****@120.46.147.143:22
要添加当前会话，点击左侧的箭头按钮。
会话管理器
所有会话
Cloud_Docker
223.80.110.43 - - [16/May/2022:09:10:48 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.2:8080"
223.80.110.43 - - [16/May/2022:09:10:49 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:49 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:10:50 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:51 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
223.80.110.43 - - [16/May/2022:09:10:51 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:52 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.4:8080"
223.80.110.43 - - [16/May/2022:09:10:53 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Win
dows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safar
i/537.36 Edg/101.0.1210.47" "-" "172.17.0.3:8080"
[root@ecs-docker logs]# cd ../
[root@ecs-docker nginx]# ls
conf logs www
[root@ecs-docker nginx]# cd conf/
[root@ecs-docker conf]# vim nginx.conf
[root@ecs-docker conf]# vim nginx.conf
[root@ecs-docker conf]# docker restart nginx
nginx
[root@ecs-docker conf]#
```

日志文件清空

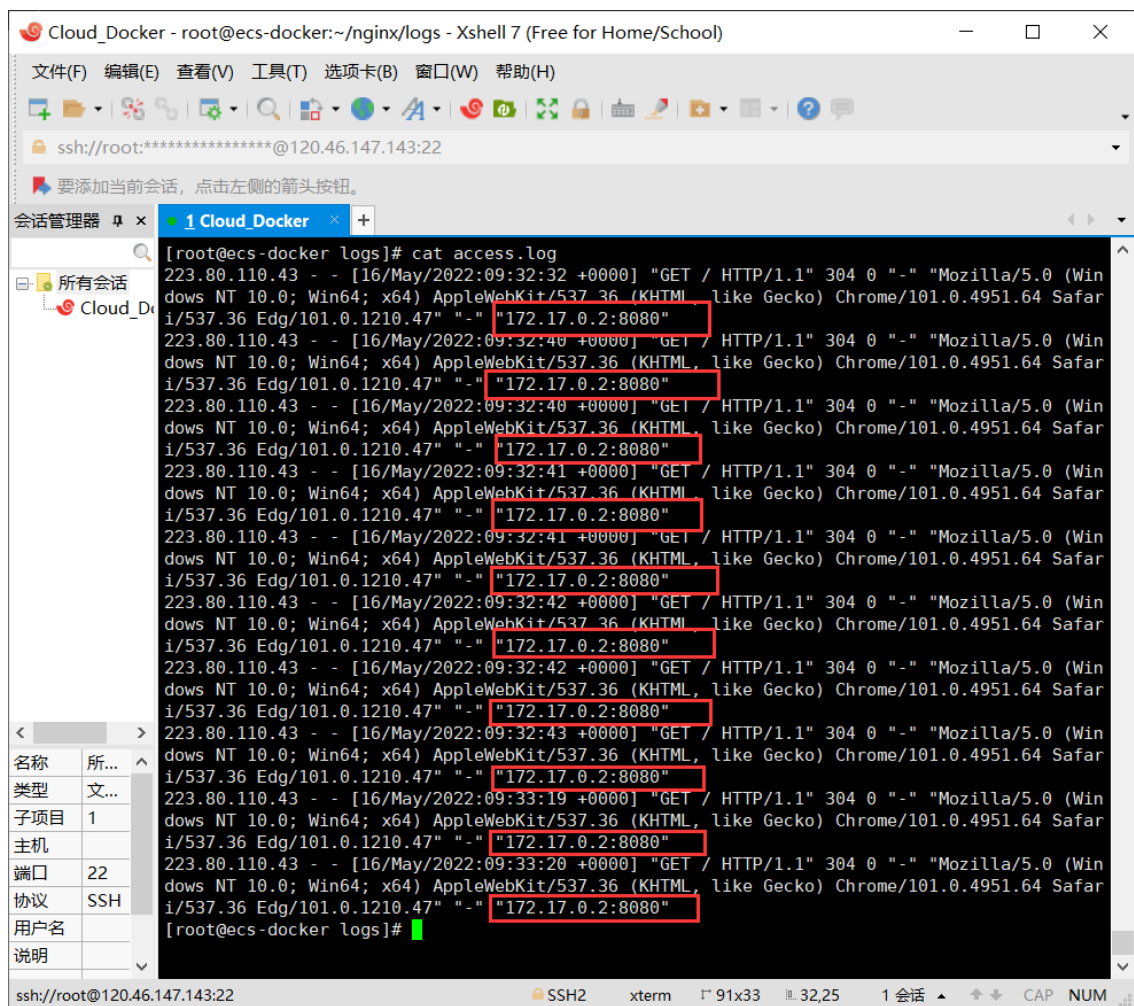
- 测试

从网页来看，每次访问时，如果主机确定，IP地址不发生变化，那么每次访问到的页面都是同一个页面。

如下所示：



从当前主机以网页访问服务器几次,查看对应的日志文件如下所示:



由上图不难看出,当物理主机IP地址不改变时,其发送的请求对应的处理服务器是一开始就已经确定了的.后续的多次访问

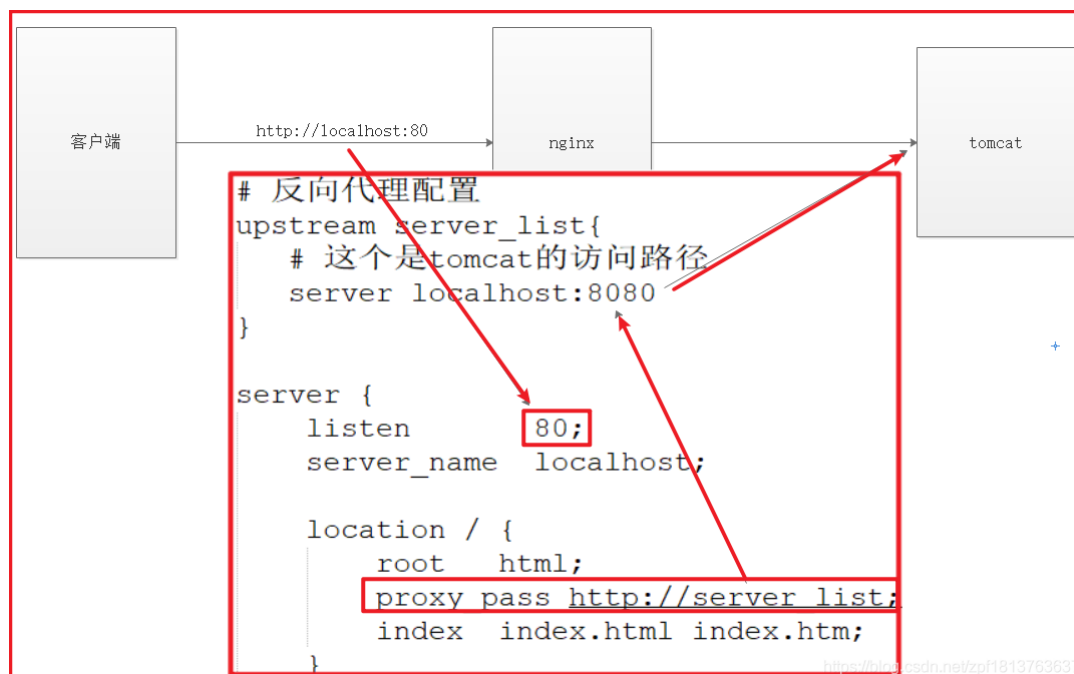
结论分析与体会

关于负载均衡与Nginx

所谓负载均衡就是：就是把大量的请求按照我们指定的方式均衡的分配给集群中的每台服务器，从而不会产生集群中大量请求只请求某一台服务器，从而使该服务器宕机的情况。

如果想要实现负载均衡,就需要在此之前实现反向代理。这里的反向代理是指，程序发送请求到某个域名，默认该请求会被Nginx获取，类似于DNS解析，Nginx会根据配置，将发送的请求转发到特定的服务器中。

实现反向代理的关键部分在于对Nginx配置文件的设置。上述过程反映到Nginx表现为下图：



其中配置文件中定义好的 `upstream server_list{}` 内部存放的是当前获取到访问请求后，可以转发到的服务器地址，服务器地址之间以 `;` 作为分隔符。

如果在对应服务器地址后添加属性 `weight: x` 代表将会执行权重分配的策略。

在整个列表的第一行添加 `ip_hash` 代表将会执行IP_HASH的分配策略。

三种负载均衡策略

轮询策略

定义：

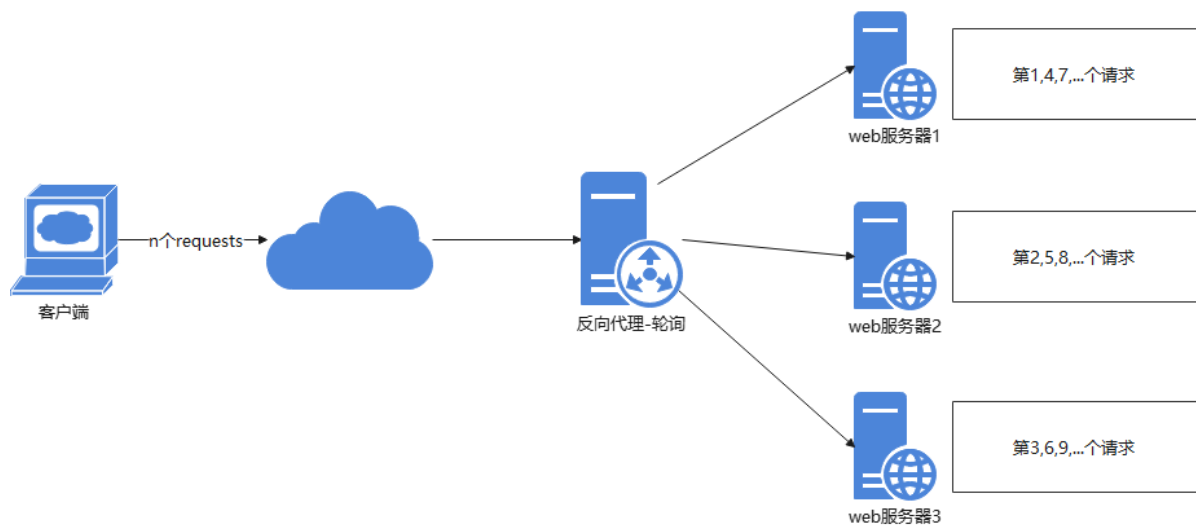
最基本的配置方法，它是upstream模块默认的负载均衡默认策略。每个请求会按时间顺序逐一分配到不同的后端服务器。含有以下参数：

属性	作用
fail_timeout	与max_fails结合使用。
max_fails	设置在fail_timeout参数设置的时间内最大失败次数，如果在这个时间内，所有针对该服务器的请求都失败了，那么认为该服务器会被认为是停机了，
fail_time	服务器会被认为停机的时间长度,默认为10s。
backup	标记该服务器为备用服务器。当主服务器停止时，请求会被发送到它这里。
down	标记服务器永久停机了。

注意点：

- 在轮询中，如果服务器down掉了，会自动剔除该服务器。
- 缺省配置就是轮询策略。
- 此策略适合服务器配置相当，无状态且短平快的服务使用。

示意图：



权重策略

定义：

权重方式，在轮询策略的基础上指定轮询的几率。

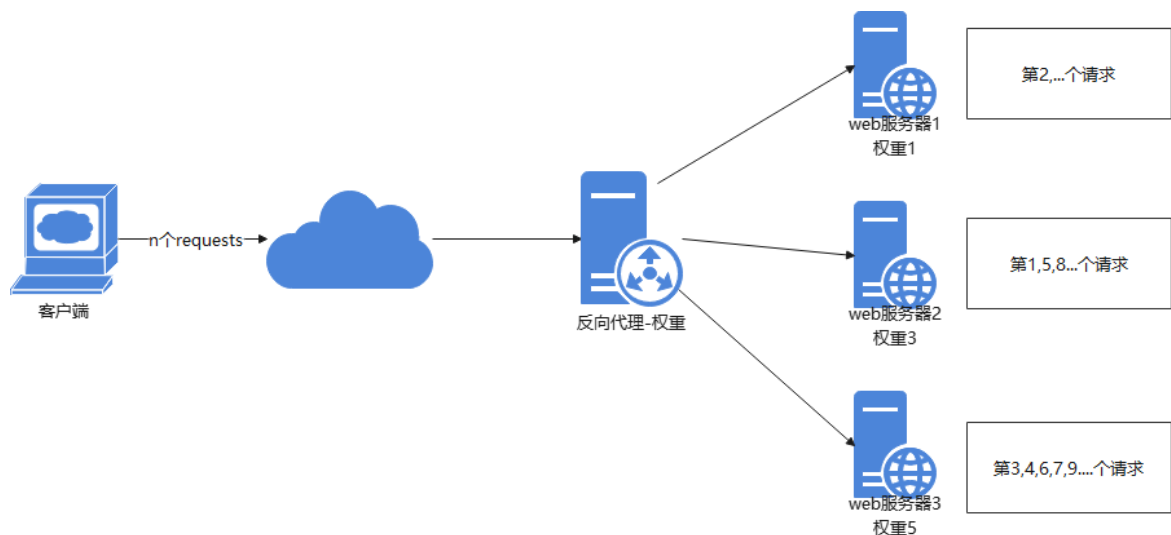
weight参数用于指定轮询几率，weight的默认值为1；

weight的数值与访问比率成正比，比如weight为2的服务器被访问的几率为weight为1的服务器的两倍。

注意点：

- 权重越高分配到需要处理的请求越多。
- 此策略可以与least_conn和ip_hash结合使用。
- 此策略比较适合服务器的**硬件配置差别比较大**的情况。硬件配置更高，可将其权重设置更大。

示意图：



注:以上请求号仅代表出现频率,不代表出现规律

IP_HASH策略

定义：

指定负载均衡器按照基于客户端IP的分配方式，这个方法确保了相同的客户端的请求一直发送到相同的服务器，以保证session会话。

每个访客都固定访问一个后端服务器，可以解决session不能跨服务器的问题。

注意点：

- 在nginx版本1.3.1之前，不能在ip_hash中使用权重（weight）。
- ip_hash不能与backup同时使用。
- 此策略适合有状态服务，比如session。
- 当有服务器需要剔除，必须手动down掉。

示意图：

