

Nginx第二讲.反向代理

[📖 播报文章](#)

修老板网络技术杂谈

2021-09-10 09:31

[关注](#)

学习nginx反向代理的功能及使用方法前, 先要了解什么是正向代理和反向代理, 以及他们的作用, 这对于接下来学习负载均衡十分必要。

一、正向代理与反向代理

1. 代理服务器

代理服务器就是位于发起请求的客户端与原始服务器端之间的一台跳板服务器, 正向代理可以隐藏客户端, 反向代理可以隐藏原始服务器。

2. 正向代理

正向代理, 意思是一个位于客户端和原始服务器(origin server)之间的服务器, 为了从原始服务器取得内容, 客户端向代理发送一个请求并指定目标(原始服务器), 然后代理向原始服务器转交请求并将获得的内容返回给客户端。

如下图1所示, 内部服务器Server-A需要与互联网上的服务器 Server-B通信, 但是A不希望把自己的内部IP地址和端口暴露到互联网, 以免产生安全隐患。因此A通过代理服务器(设备)Proxy-C, 与互联网上的服务器B进行通信。

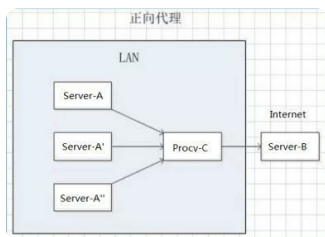


图1-正向代理

3. 反向代理

反向代理的工作原理是, 代理服务器来接受客户端的网络访问连接请求, 然后服务器将请求有策略的转发给网络中实际工作的业务服务器, 并将从业务服务器处理的结果, 返回给网络上发起连接请求的客户端。

反向代理有以下优点:

- (1) 提高了内部服务器的安全;
- (2) 加快了对内部服务器的访问速度;
- (3) 节约了有限的IP资源。

如下图2所示, 客户端Client-A想要访问www.tonglaoban.com, 访问请求通过地址解析后, 发送到IP地址为1.1.1.1:80的代理服务器, 通过代理服务器的正向代理以及负载均衡功能, 最终的访问请求通过轮询(暂时只讨论轮询的负载均衡)的方式分发到Server-A、Server-B、Server-C。

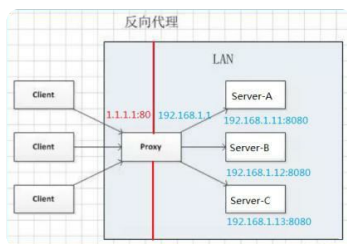


图2-反向代理

二、实验环境

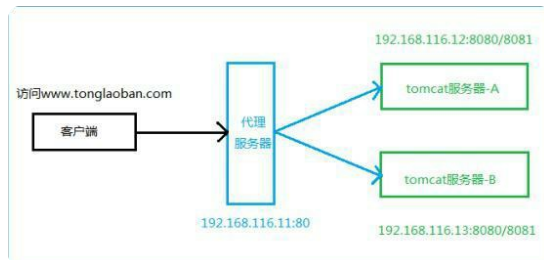


图3-实验环境

三、配置实例

1. 实例1-反向代理基本配置

1. 实现效果

相关推荐



搭建企业网站, 搭建网站的步骤介绍

科技



Java编程中的MVC设计模式

李兴华java2pyt...



Redis 事务使用不当引发的生产事故

JB 2评论



什么是CDN? CDN的技术里是什么?

三方



基于java学生竞赛管理系统

博源阅读文化社



Java编程中的MVC设计模式

李兴华java2pyt...



Redis 事务使用不当引发的生产事故

JB 2评论



什么是CDN? CDN的技术里是什么?

三方



基于java学生竞赛管理系统

博源阅读文化社

百度热搜

[换一换](#)

- 1 新华社:一封了之、一放了之...
- 2 31省份昨日新增本土2276+2...
- 3 携手应对挑战 加快经济复苏
- 4 张伯礼:奥密戎难以彻底消灭
- 5 伊藤美诚11比0
- 6 长江网遣非法捕捞嫌疑人:吃了3条
- 7 “椰树四美”穿羽绒服直播跳操
- 8 顾客吐槽烤全羊烤糊遭索赔2万
- 9 澳大利亚为何急于改善对华关系?
- 10 香港外籍法官“放生”毒贩后请辞

打开浏览器，输入www.tonglaoban.com，通过nginx代理服务器(192.168.116.11)的80端口，跳转到真是tomcat服务器(192.168.116.12)的8080端口。

2.准备工作

(1)在linux系统中安装tomcat服务，使用默认端口8080。

(2)安装tomcat之前需要检查linux服务器的java环境。

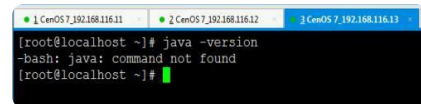


图4-实验过程

如服务器没有安装java环境，需要先安装，本次以通过yum源安装的方式作为示例，该安装方式可以直接将其他依赖文件一并安装：



图5-实验过程

安装成功后，再次查看java版本：

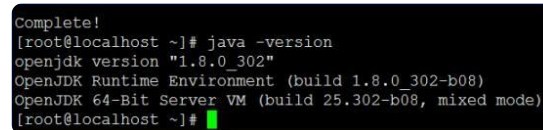


图6-实验过程

(3)安装tomcat，Tomcat安装文件放到linux系统中，解压/usr/src/apache-tomcat。



图7-实验过程

(4)进入tomcat的bin目录中，./start.sh启动，进入log目录中，查看日志。



图8-实验过程

(5)打开网页查看状态http://192.168.116.13:8080/

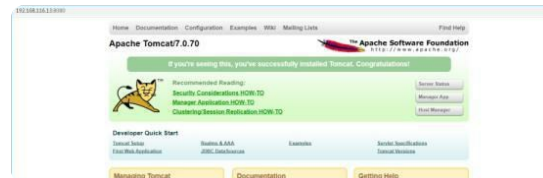


图9-实验过程

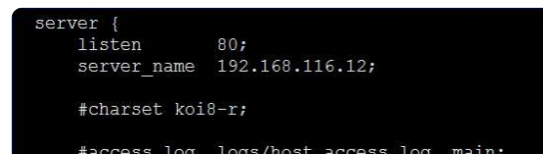
3.反向代理配置

(1)配置域名解析

(2)在以下路径C:\Windows\System32\drivers\etc找到host文件，并增加内容：

(3)192.168.116.11 www.tonglaoban.com

(4)在nginx进行请求转发的配置(反向代理配置)，配置文件在/usr/local/nginx/conf，编辑nginx.conf：



```
location / {
    root   html;
    proxy_pass http://192.168.116.12:8080;
    index index.html index.htm;
}
```

图10-实验过程

修改配置文件，并保存好，切记要重启nginx服务。

```
[root@localhost nginx]# cd sbin/
[root@localhost sbin]# ls
nginx
[root@localhost sbin]# ./nginx -s reload
[root@localhost sbin]#
```

图11-实验过程

结果如下：

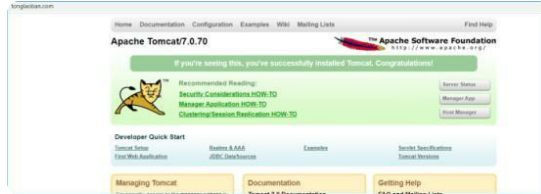


图12-实验过程

(5) 如果希望更直观的看到测试效果，可以在tomcat服务器增加测试页面，在/usr/src/apache-tomcat-7.0.70/webapps路径中，新建test112文件夹，文件夹中新建test112.html文件：

```
[root@localhost test112]# pwd
/usr/src/apache-tomcat-7.0.70/webapps/test112
[root@localhost test112]# vi test112.html
<h1>This is 192.168.116.12 </h1>
我是CCIE_佟老板，一个让你沉溺于网络学习世界的好伙伴！
```

图13-实验过程

效果如下：



图14-实验过程

2. 实例2-根据不同uri分配到不同服务器

1. 实现效果：

根据不同uri分配到不同服务器(或端口)，监听9001端口

http://192.168.116.11:9001/8080分配到192.168.116.12:8080

http://192.168.116.11:9001/8081分配到192.168.116.13:8081

2. 准备工作

(1) 准备两个tomcat服务器，一个8080端口，一个8081端口。

在/usr/src/apache-tomcat-7.0.70/conf路径，编辑server.xml文件：

```
<!-- A "Connector" represents an endpoint by which requests
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html (blocking
ing)
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL HTTP/1.1 Connector on port 8081
-->
<Connector port="8081" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
port="8081" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
-->
```

图15-实验过程

新建test文件夹，文件夹中新建test.html文件，如实例1填写IP地址。



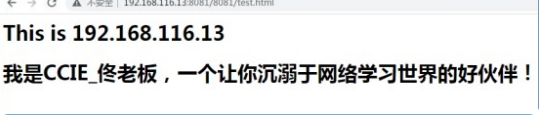


图16-实验过程

3.配置nginx代理服务

对外监听端口为9001, 根据不同uri, 分配到不同应用tomcat服务器。

http://192.168.116.11:9001/8080分配到192.168.116.12:8080

http://192.168.116.11:9001/8081分配到192.168.116.13:8081

在nginx配置文件中, 新建server块配置如下:

```
server {
    listen      9001;
    server_name 192.168.116.11;

    location ~ /8080/ {
        proxy_pass http://192.168.116.12:8080;
    }

    location ~ /8081/ {
        proxy_pass http://192.168.116.13:8081;
    }
}
```

图17-实验过程

重启nginx服务。

```
[root@localhost nginx]# cd sbin/
[root@localhost sbin]# ls
nginx
[root@localhost sbin]# ./nginx -s reload
[root@localhost sbin]#
```

图18-实验过程

4.测试结果

访问http://192.168.116.11:9001/8080/test.html



图19-实验过程

访问http://192.168.116.11:9001/8081/test.html



图20-实验过程



[举报/反馈](#)

大家都在搜

- NGINX反向代理
- NGINX转发
- web反向代理
- 反向代理解决跨域的原理
- ssh 反向代理
- 反向代理

发表评论

发表神评论

发表

评论列表(1条)

- 0无爱的小家0
可以 😊
10-24 17:20 江苏

没有更多啦

