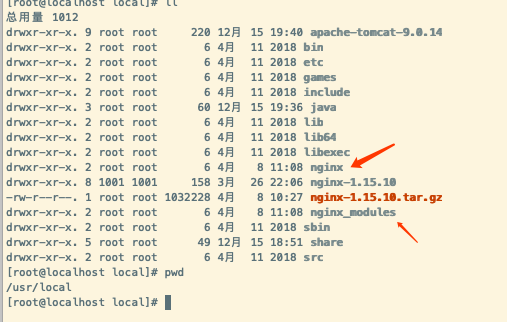
**Nginx+keepalived**

# nginx的下载

# 安装nginx之前先安装依赖



# nginx的安装

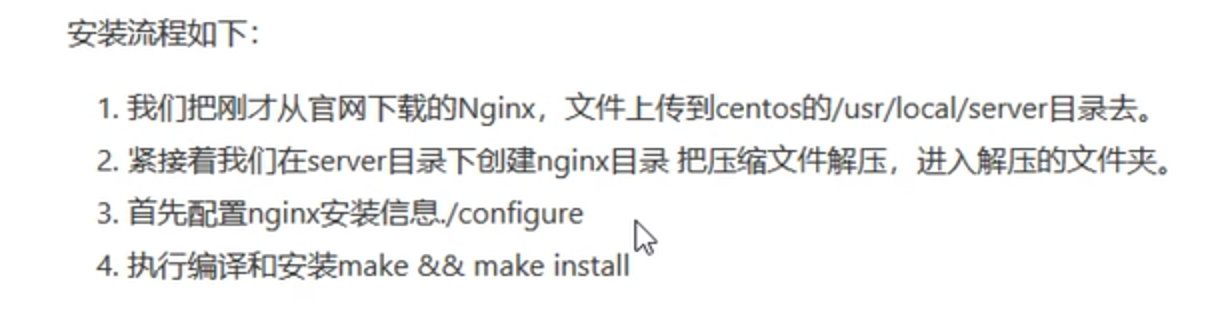


目录介绍：

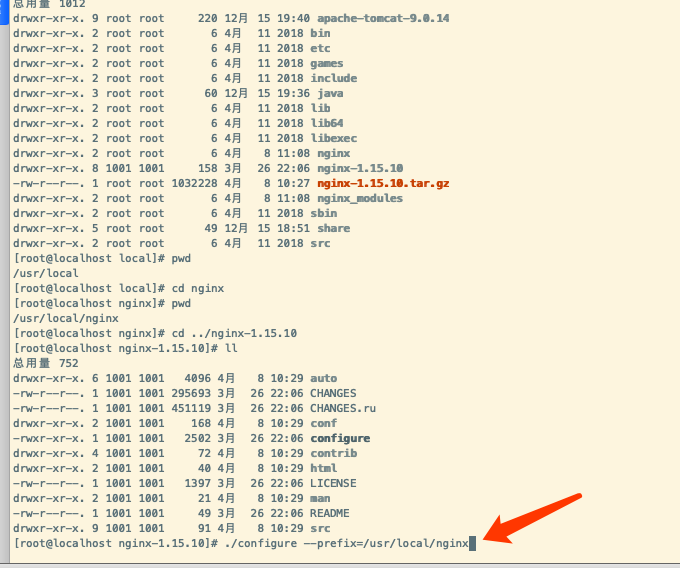
1，nginx目录为正式安装的文件目录

2，nginx-1.15.10为解压目录，并没有正式安装

3，nginx\_modules为第三方模块目录



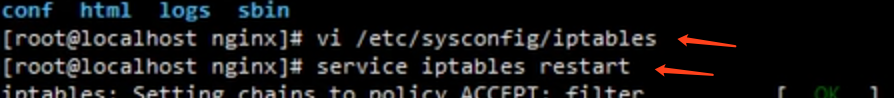
本机实际与视频演示目录地址不一样



进行安装[root@localhost nginx-1.15.10]# ./configure --prefix=/usr/local/nginx  
添加对ssl证书的支持：./configure --with-http\_ssl\_module --prefix=/usr/local/nginx

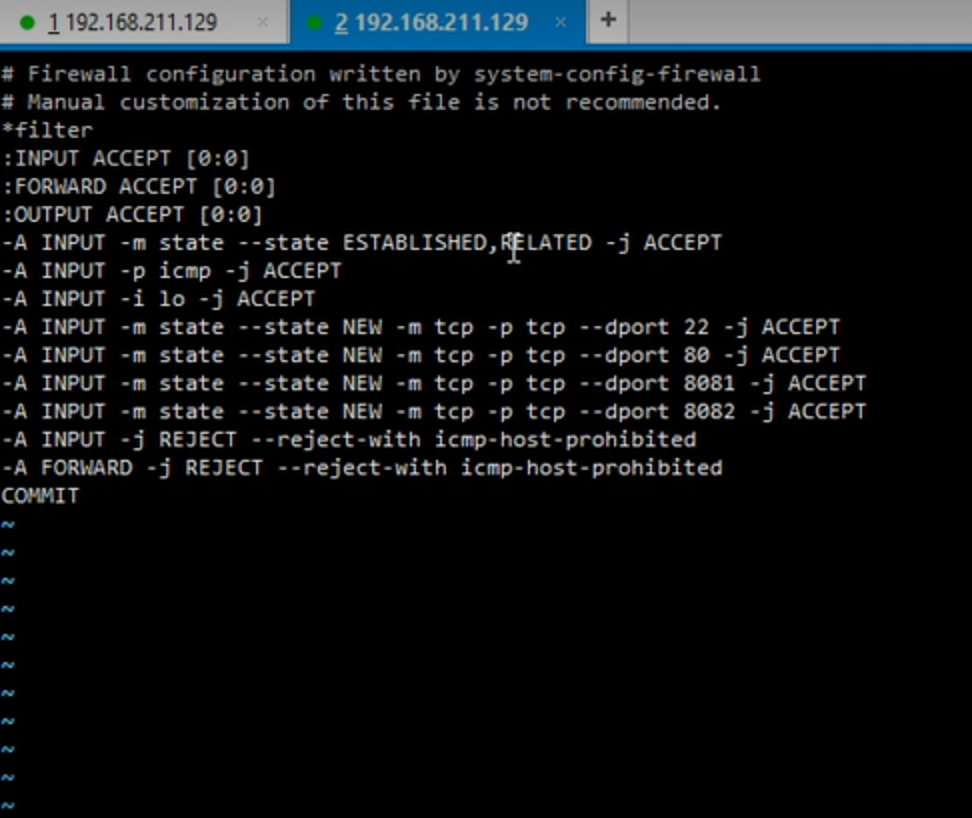
继续执行命令：

防火墙开放端口：



命令一：打开防火墙端口控制

命令二：重启防火墙



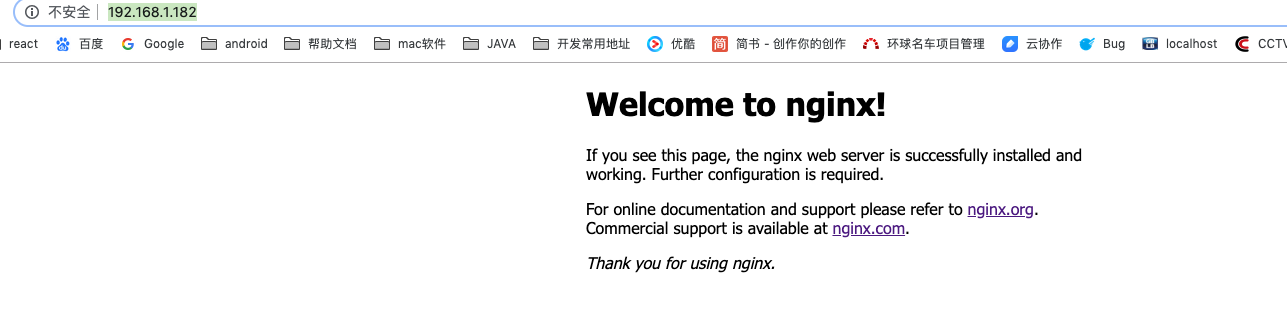
防火墙端口开发编写。

# nginx常用指令



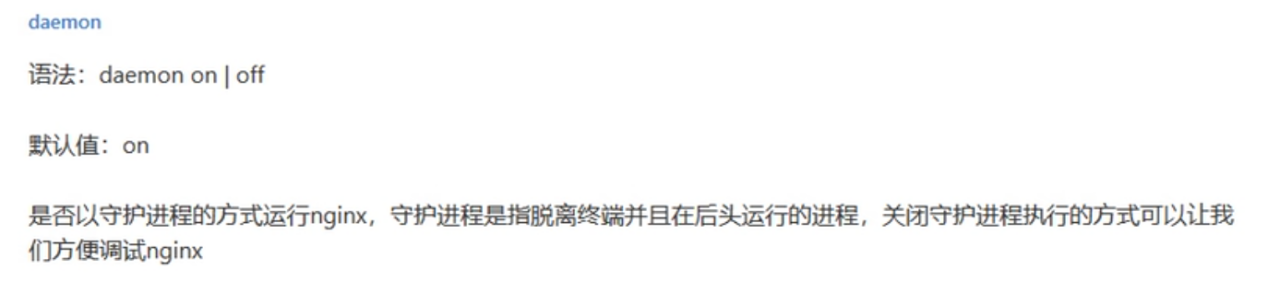
启动：进入nginx下的sbin文件夹，执行。/nginx命令即可

使用ip进行访问，出现如下即代表成功：

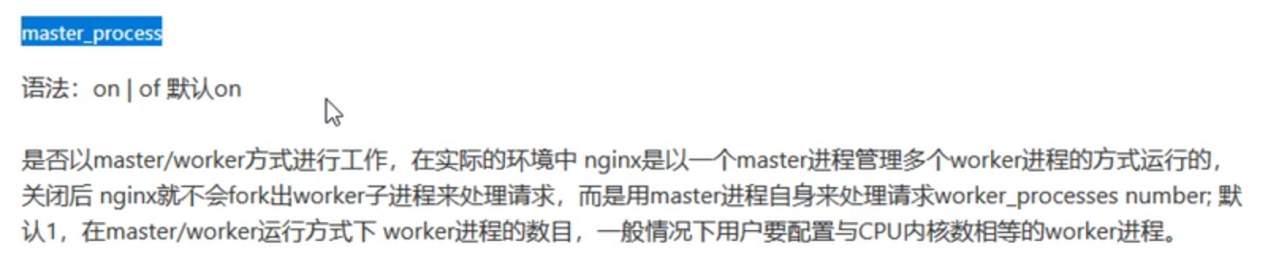


# 主模块

1，是否在后台运行



2，是否多现场运行



3，local规则：

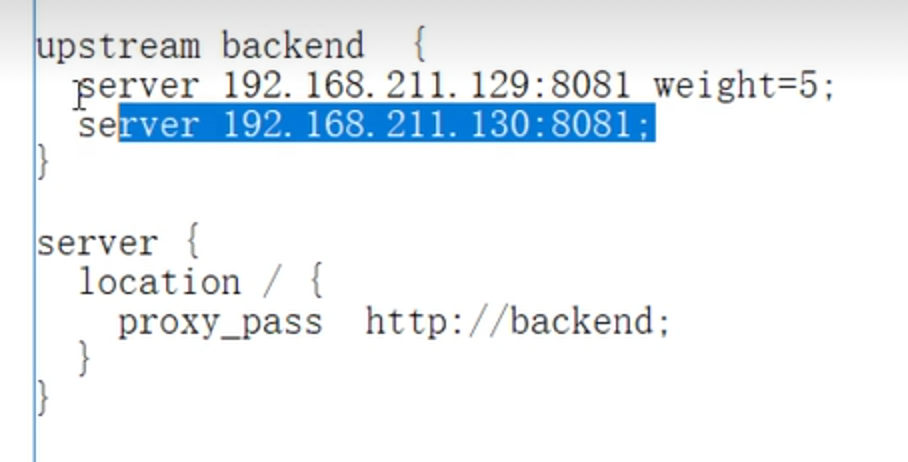


# http负载均衡模块



负载均衡池的概念。

可以写成如下：

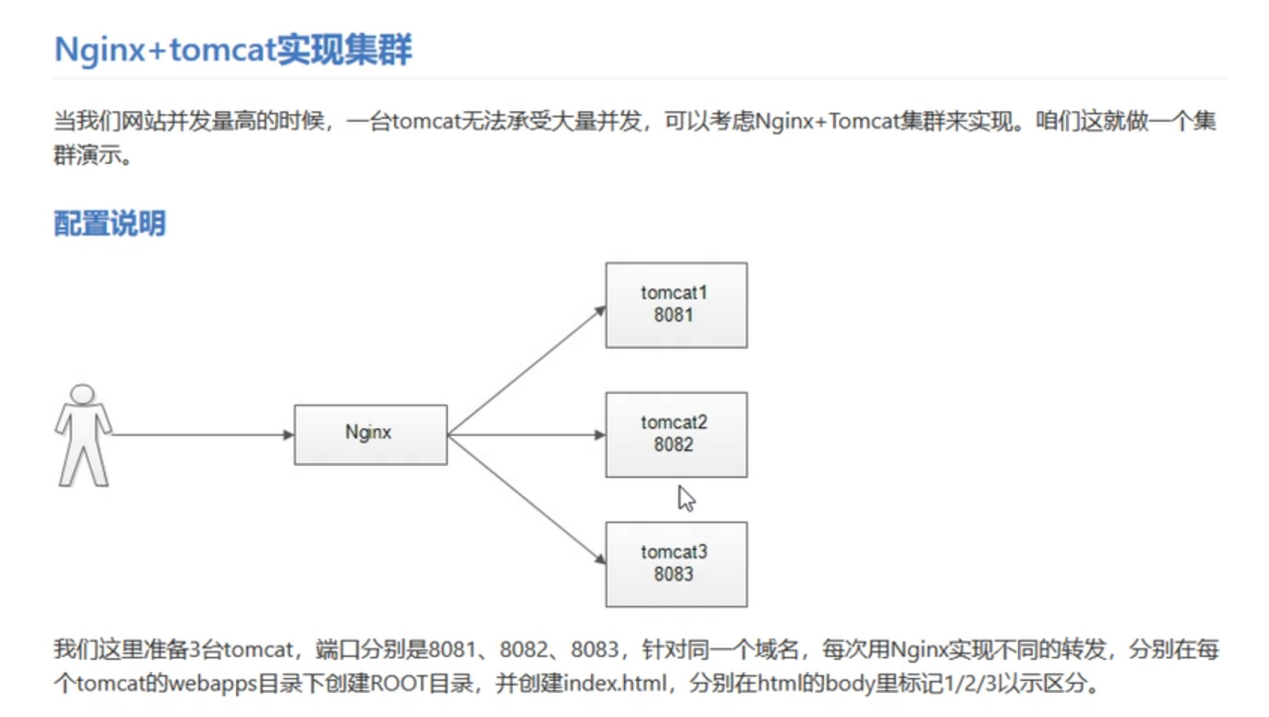


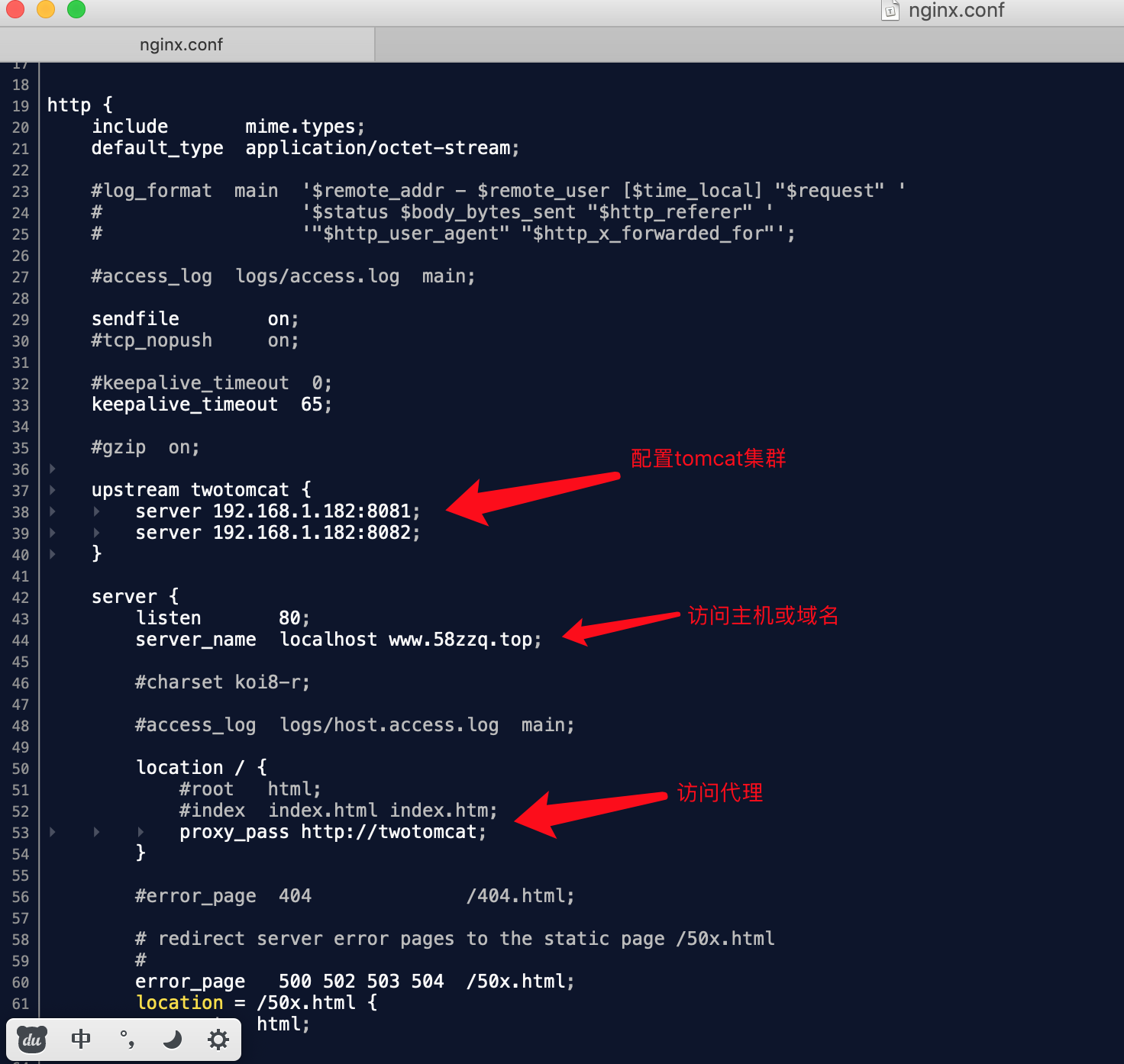
所有的请求转发给两台机器进行处理。

# 访问限制

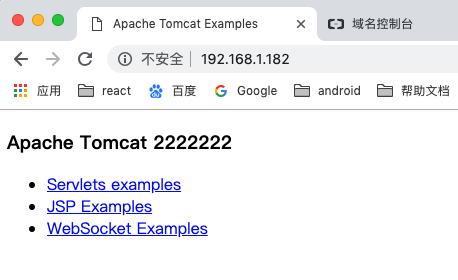
deny禁止访问 denyall全部禁止

# 八、nginx+tomcat实现集群



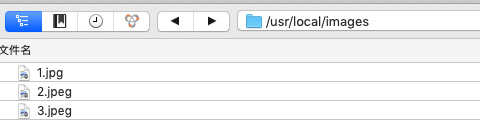


如上示例进行配置，访问如下：或访问：www.58zzq.top

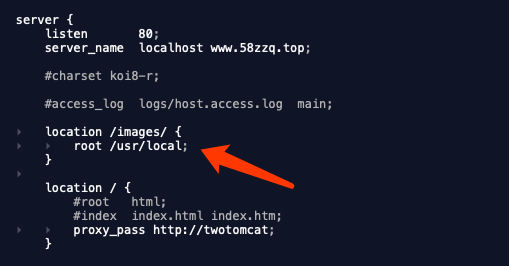


# nginx搭建图片服务器

1,在虚拟机上新建images文件夹并存入图片



2，配置拦截的请求 【地址里不要加入images】

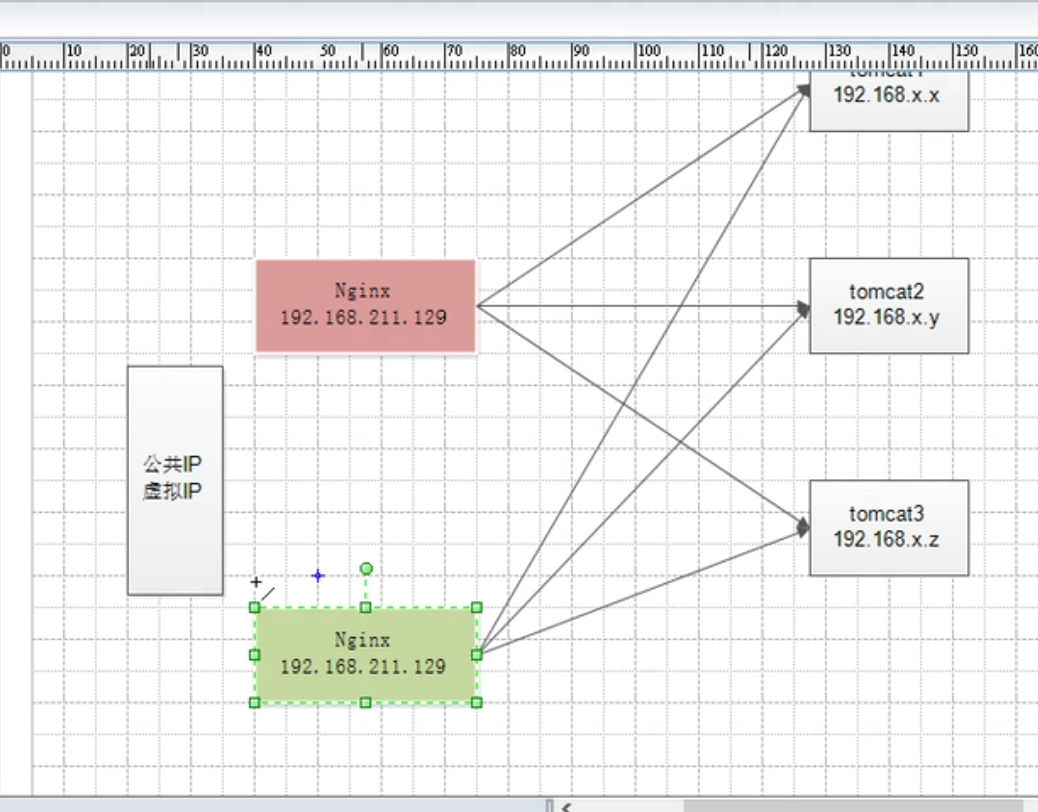


3，进行网络请求

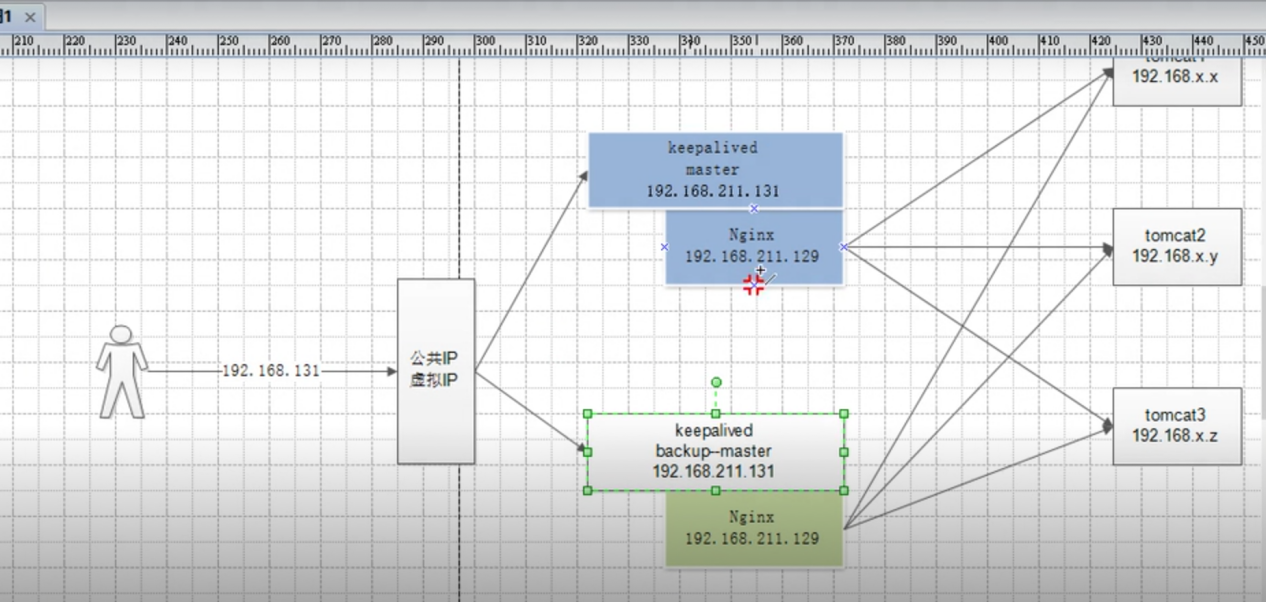


网络请求成功.

# keepalived+nginx



搭建两个nginx服务器，两个nginx服务器都配置多个tomcat的集群模式，最外层弄一个虚拟ip进行访问。



keepalived相当于路由，与对应的nginx进行绑定。

# keepalived安装

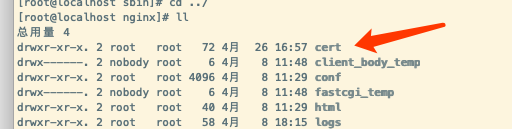






# nginx安装ssl正式及配置

到域名管理中心下载ssl正式：一个pen文件、一个key文件

然后放在nginx文件下：

我这里放在nginx下的cert文件夹下（自己建的文件夹）

配置nginx.conf文件，如下：

|  |
| --- |
| #user nobody;  worker\_processes 1;  #error\_log logs/error.log;  #error\_log logs/error.log notice;  #error\_log logs/error.log info;  #pid logs/nginx.pid;  #daemon off/on是否后台运行  events {  worker\_connections 1024;  }  http {  include 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"';  #access\_log logs/access.log main;  sendfile on;  #tcp\_nopush on;  #keepalive\_timeout 0;  keepalive\_timeout 65;  #gzip on;    upstream twotomcat {  server 192.168.1.182:8081;  server 192.168.1.182:8082;  }  # another virtual host using mix of IP-, name-, and port-based configuration  #  #server {  # listen 8000;  # listen somename:8080;  # server\_name somename alias another.alias;  # location / {  # root html;  # index index.html index.htm;  # }  #}  # HTTPS server  #  server {  listen 443 ssl;  server\_name localhost www.58zzq.top;  #ssl on;  ssl\_certificate ../cert/1108646\_www.58zzq.top.pem;  ssl\_certificate\_key ../cert/1108646\_www.58zzq.top.key;  ssl\_session\_cache shared:SSL:1m;  ssl\_session\_timeout 5m;  ssl\_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;  ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;  ssl\_prefer\_server\_ciphers on;  location / {  # root html;  # index index.html index.htm;  proxy\_pass http://twotomcat;  }  }  server {  listen 80;  listen [::]:80;  server\_name www.58zzq.top;  return 302 https://$server\_name$request\_uri;  }  } |

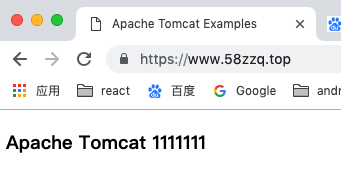
查看配置文件是否生效，如果为success则成功，报错需要修改。

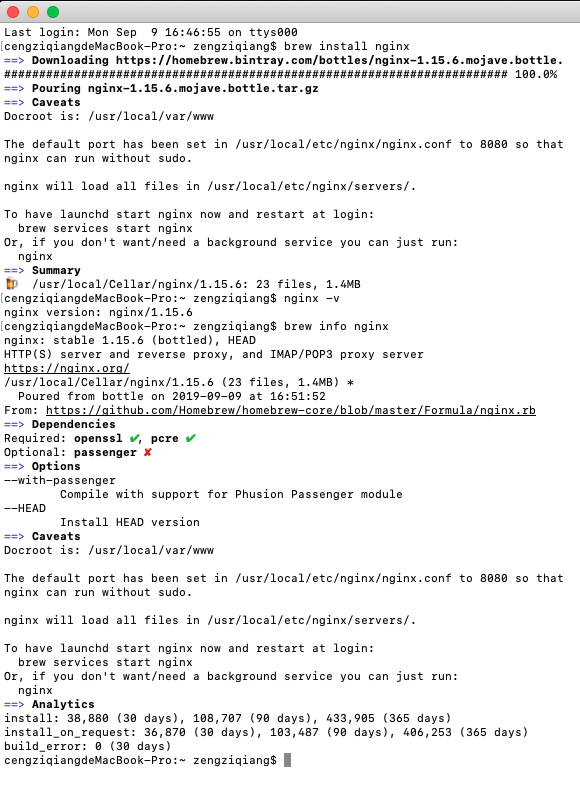
|  |
| --- |
| $ nginx -t // 检查nginx配置文件 |

重启nginx

|  |
| --- |
| $ nginx -s reload // 使配置生效 |

访问你的域名，如下则表示成功。





# nfc实现文件共享

Linux1：192.168.1.182

Linux2：192.168.1.249，有Nginx的服务器

两台机器都需要安装nfc和rpcbind

检查是否安装：rpm -qa|grep nfs

rpm -qa|grep rpcbind

安装命令：

yum -y install rpcbind nfs-utils

启动：

systemctl start rpcbind

systemctl start nfs

查看启动状态是否成功：

systemctl status nfs

systemctl status rpcbind

这里与 【/usr/local/images】文件夹为例

给权限：chmod 777 /usr/local/images

现在去Linux1上去操作，将Linux1上/usr/local/images文件夹里的文件共享到Linux2里，这样Linux2上的Nginx就可以访问到了。

【重点】在Linux1上操作：

vi /etc/exports

输入：/usr/local/images \*(rw)

/usr/local/images \*(insecure,rw,async,no\_root\_squash)

保存，重启使生效：exportfs -r

【重点】在Linux2上操作挂起：

mount -t nfs 192.168.1.182:/usr/local/images /usr/local/images

查看是否挂起成功：df

两个操作别弄反了 【<https://wenku.baidu.com/view/ffe239c2aa00b52acfc7ca36.html>

】