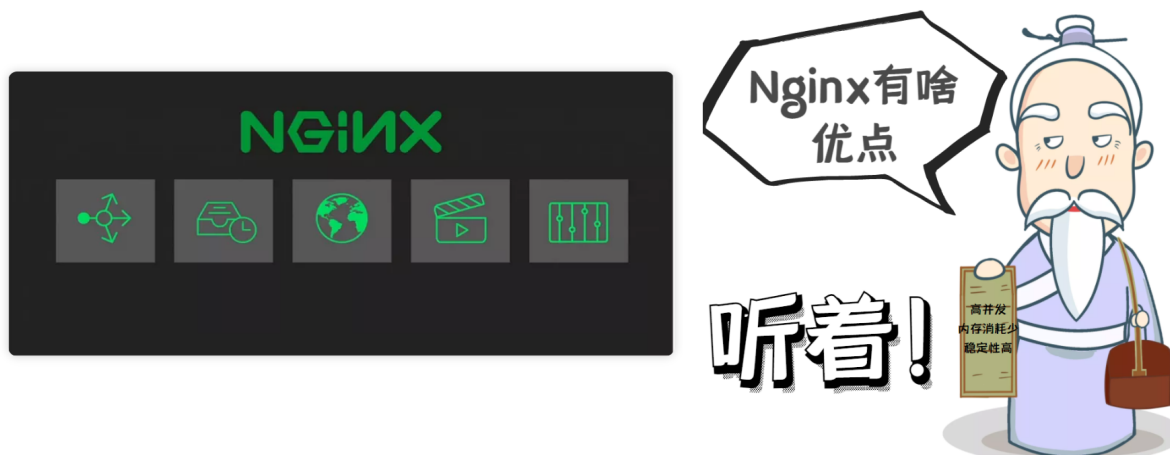


## 【第59话：Nginx那些可能被问到的问题】

Hello 小伙伴们，这节课给大家讲一下"你的项目中哪里使用了Nginx"。

我们先来说说什么是Nginx

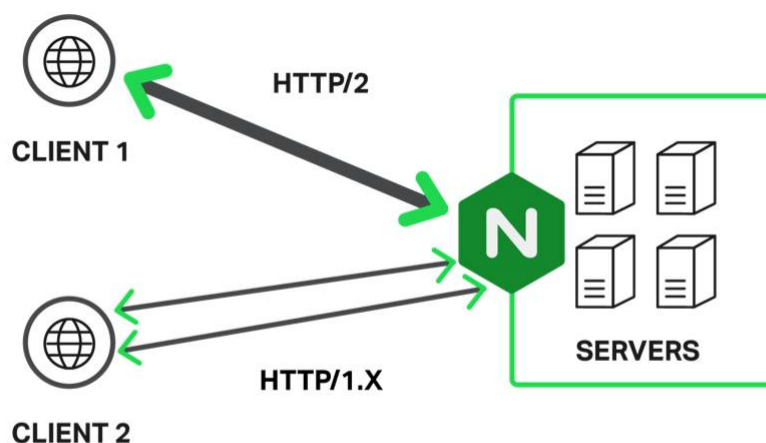


Nginx是一款轻量级的Web服务器、反向代理服务器，由于它的内存占用少，启动极快，高并发能力强，在互联网项目中广泛应用。Nginx 专为性能优化而开发，使用异步非阻塞事件驱动模型。

## Nginx 四大应用场景

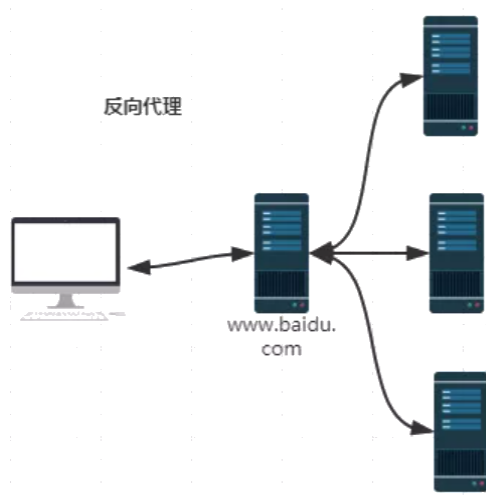
### HTTP服务器

Nginx本身也是一个静态资源的服务器，当只有静态资源的时候，就可以使用Nginx来做服务器，如果一个网站只是静态页面的话，那么就可以通过这种方式来实现部署。



## 反向代理

反向代理应该是Nginx使用最多的功能了，反向代理(Reverse Proxy)方式是指以代理服务器来接受internet上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给internet上请求连接的客户端，此时代理服务器对外就表现为一个反向代理服务器。



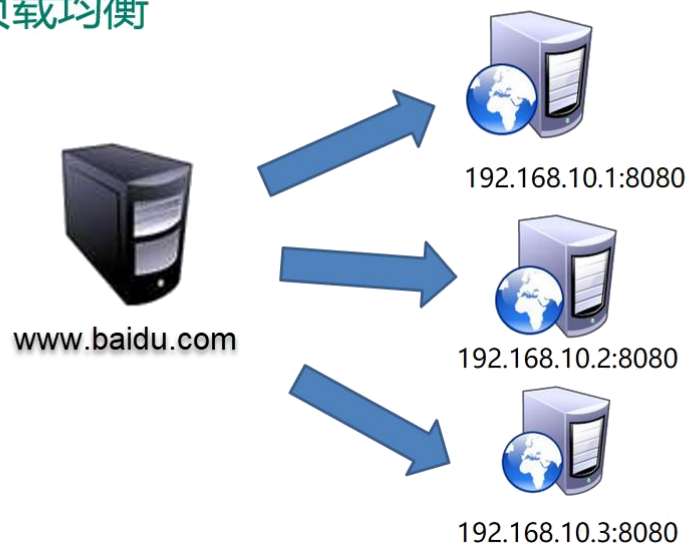
配置内容

```
server{
    ....
    listen 80;
    server_name suiyi;
    location / {
        proxy_pass http://192.168.0.106:8080;
    }
}
```

## 负载均衡

负载均衡也是Nginx常用的一个功能，负载均衡其意思就是分摊到多个操作单元上进行执行，例如Web服务器、FTP服务器、企业关键应用服务器和其它关键任务服务器等，从而共同完成工作任务。

### 负载均衡



## 语法格式:

```
upstream bjsxt {                # upstream 需要一个名字
    server 192.168.0.106:8080 max_conns=2 fail_timeout=1s;
    server 192.168.0.106:8081 weight = 1;
    server 192.168.0.106:8082 weight = 1;
}
server{
    listen 80;
    server_name suiyi;
    location / {
        proxy_set_header Host $host; #转发原始请求的host头部
        proxy_pass http://bjsxt;    #转发到upstream块定义的服务器集群。
    }
}
```

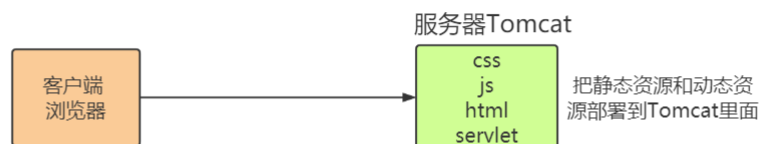
### upstream参数:

- max\_conns 节点的最大连接数
- slow\_start 缓慢启动时间
- down 节点下线
- backup 备用节点
- max\_fails 允许的最大失败数
- fail\_timeout 超过最大失败数后的等待时间

## 动静分离

动静分离是让动态网站里的动态网页根据一定规则把不变的资源和经常变的资源区分开来，动静资源做好了拆分以后，我们就可以根据静态资源的特点将其做缓存操作，这就是网站静态化处理的核心思路。

动静分离之前



动静分离之后

