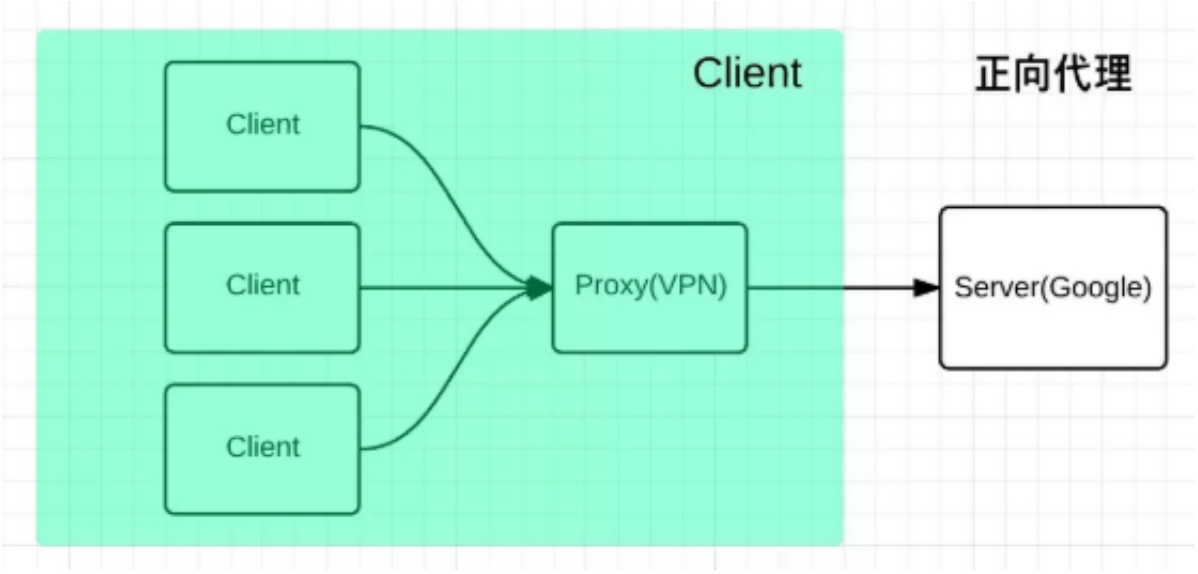
前端开发需要了解的Nginx知识

Nginx概念

> nginx是一款自由的、开源的、高性能的HTTP服务器和**反向代理**服务器；
> 同时也是一个IMAP、POP3、SMTP代理服务器；
> nginx可以作为一个HTTP服务器进行网站的发布处理；
> 另外nginx可以作为反向代理进行负载均衡的实现。

正向代理和反向代理

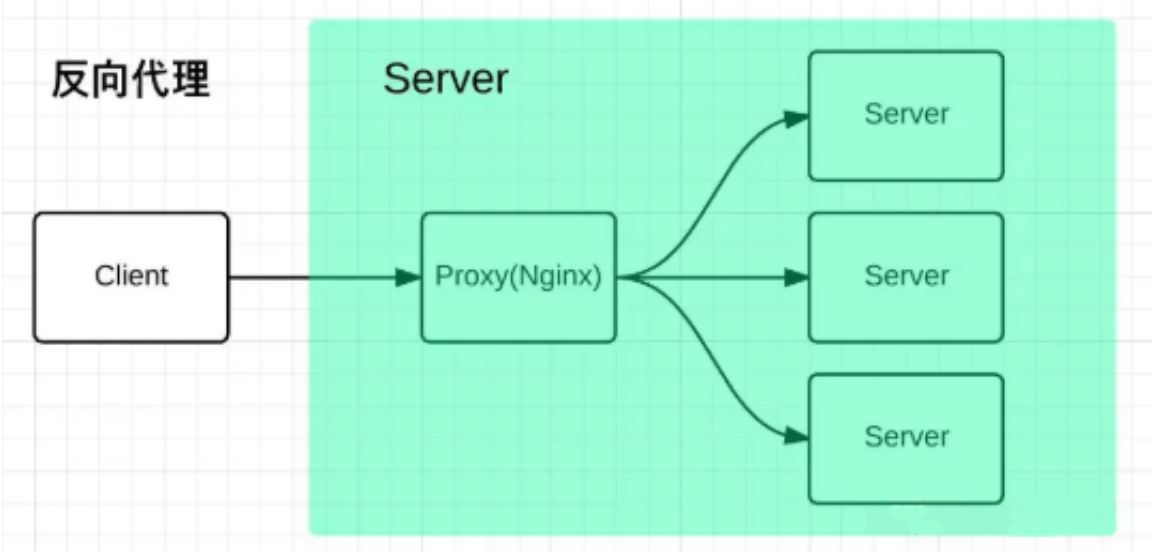
正向代理是一个位于客户端和原始服务器之间的服务器，为了从原始服务器取得内容，客户端向代理发送一个请求并指定目标(原始服务器)，然后代理向原始服务器转交请求并将获得的内容返回给客户端。
正向代理是为我们服务的，即为客户端服务的，客户端可以根据正向代理访问到它本身无法访问到的服务器资源。
正向代理对我们是透明的，对服务端是非透明的，即服务端并不知道自己收到的是来自代理的访问还是来自真实客户端的访问。



反向代理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。

反向代理是为服务端服务的，反向代理可以帮助服务器接收来自客户端的请求，帮助服务器做请求转发，负载均衡等。

反向代理对服务端是透明的，对我们是非透明的，即我们并不知道自己访问的是代理服务器，而服务器知道反向代理在为他服务。



解决跨域

跨域：同源策略限制了从同一个源加载的文档或脚本如何与来自另一个源的资源进行交互。这是一个用于隔离潜在恶意文件的重要安全机制。通常不允许不同源间的读操作。

同源：如果两个页面的协议，端口（如果有指定）和域名都相同，则两个页面具有相同的源。

```
server {
    listen  85;
    server_name  localhost;


    location /migu_bvreport{
        proxy_pass https://172.20.78.93/migu_bvreport;
    }
}
```

以上代码在访问以 http://localhost:85/migu_bvreport开头的请求时，请求会被代理去访问 https://172.20.78.93/migu_bvreport开头的请求，这样可以绕开浏览器的同源策略

自定义错误页面和访问设置

```
error_page  500 502 503 504  /50x.html;
error_page  404 http://xxx.cn
```

在nginx配置中找到第一段代码，这里定义了出现四种500错误的默认转向地址网站根目录下的50x.html。
还可以把错误页面转向其他站点，第二段代码将404错误转向到了其他站点

```
location / {
    deny   10.167.6.40;
    allow  192.168.22.20;
}
```

以上代码简单的实现了权限访问控制

gzip

gzip是网页的一种网页压缩技术，经过gzip压缩后，页面大小可以变为原来的30%甚至更小。更小的网页会让用户浏览的体验更好，速度更快。gzip网页压缩的实现需要浏览器和服务器的支持
当浏览器支持gzip压缩时，会在请求消息中包含 Accept-Encoding:gzip，这样nginx就会向浏览器发送经过gzip压缩的内容，同时在相应的信息头加入 Content-Encoding:gzip,声明这是gzip后的内容，浏览器需要先解压才可以解析解出。

```
http {
    gzip on;
```

```
    gzip_types text/plain application/javascript text/css;
}
```

以上是nginx对gzip的简单配置，这样就对html,javascript,css完成了gzip压缩

客户端适配

如今很多网站都存在PC站和mobile站两个站点，因此根据用户的浏览环境自动切换站点是很常见的需求。nginx可以通过内置变量**$http_user_agent**，获取到请求客户端的userAgent，从而知道用户处于移动端还是PC，进而控制重定向到mobile站还是PC站

```
server {
    listen 8001;
    server_name localhost;
    location / {
      # 默认的PC访问路径
      root /html/sso;
      # 当客户端是以下移动设备时，进入不同的路径
      if ($http_user_agent ~* '(Android|webOS|iPhone|iPod|BlackBerr
y)') {
          root /html/sso-mobile;
      }
      index index.html;
    }
}
```

多个虚拟主机

nginx 可以通过配置虚拟主机把多个不同域名的网站部署在同一台服务器上，即解决了维护服务器技术的难题，同时节省了服务器硬件成本和相关的维护费用。
虚拟主机配置可以基于端口号、基于IP和基于域名。

```
http {
    ...
    # 虚拟主机引入
    include vhost/*.conf;
    ...
}
```

```
# redsea.conf
```

```
# 通过不同端口号创建的虚拟主机
server  {
  listen 8085;
  server_name localhost;
  root html/redsea;
  index index.html;
  try_files  $uri  $uri/  /index.html;
  ...
}
```

配置https

微信小程序如今非常流行，大批前端开发都会涉足微信小程序，但是微信小程序只支持https请求，nginx配置https服务流程：

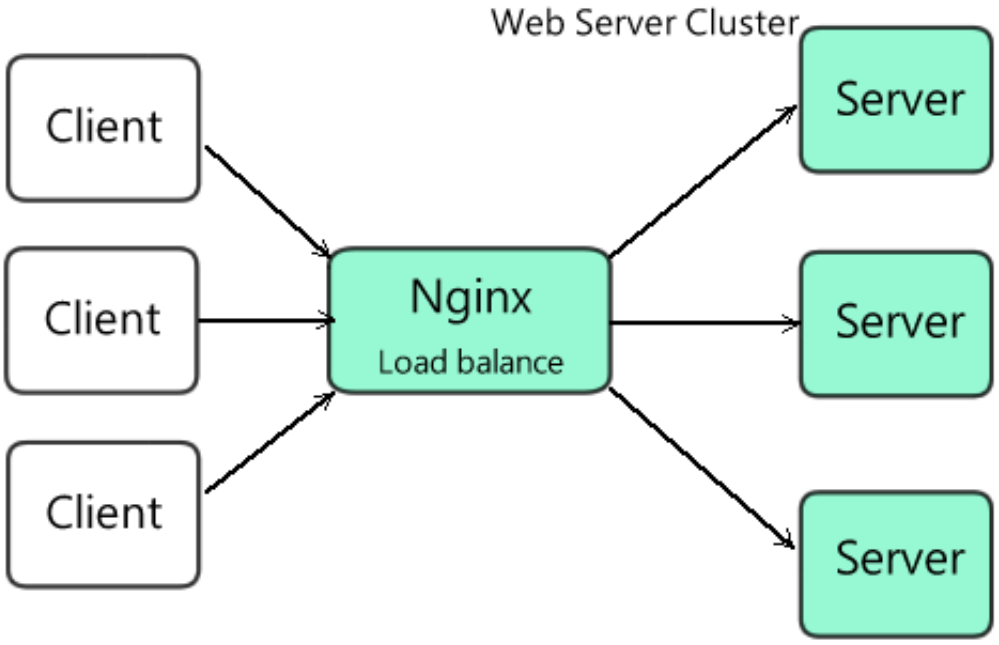
- 申请证书，便宜ssl可以申请三个月免费ssl证书，申请过程请按照官方提示步骤
- 申请完成后下载nginx版本的证书至本地，一个crt文件一个key文件，crt为证书，key为私钥
- 配置nginx

```
server {
    # 监听443端口，因为443端口是https的默认端口。80为http的默认端口
    listen  443 ssl;
    server_name  localhost;
    # 证书(公钥.发送到客户端的)
    ssl_certificate      cert.crt;
    # 私钥
    ssl_certificate_key  cert.key;

    location / {
        root   html;
        index  index.html index.htm;
    }
}
```

负载均衡

网站的访问量越来越大，服务器的服务模式也得进行相应的升级，比如分离出数据库服务器、分离出图片作为单独服务，这些是简单的数据的负载均衡，将压力分散到不同的机器上。nginx能将同一个域名的访问分散到两台或更多的机器上，缓解web前端的压力，达到负载均衡的目的



nginx负载均衡配置,主要是proxy_pass,upstream的使用，nginx 的 upstream支持以下种方式的分配

- 轮询（默认）
  每个请求按时间顺序逐一分配到不同的后端服务器，如果后端服务器down掉，能自动剔除。
- weight
  指定轮询几率，weight和访问比率成正比，用于后端服务器性能不均的情况。
- ip_hash
  每个请求按访问ip的hash结果分配，这样每个访客固定访问一个后端服务器，可以解决session的问题。
- fair（第三方）
  按后端服务器的响应时间来分配请求，响应时间短的优先分配。
- url_hash（第三方）

```
http {
    upstream www.foo.com {
      server   10.146.30.161:8080;
      server   10.146.30.162:8080;
    }

    server {
        ...
        location /migunet_cloudbi {
             root   /;
             proxy_pass   http://www.foo.com;
           }
        ...
    }
```

```
    }
```

nginx负载均衡依赖于 ngx_http_upstream_module 模块，ngx_http_upstream_module 允许nginx定义一组或多组节点服务器组，使用时可以通过 proxy_pass代理方式把网站的请求放送到事先定义好的对应upstream组中的服务器。