Nginx

Nginx安装

```
1  #单体
2  docker run -d -p 8090:80  -v
   /home/nginx/:/var/log/nginx/ \
3  -v
   /home/nginx/config/nginx.conf:/etc/nginx/nginx.con
   f --name nginx  nginx
4  #单体  监听多个端口号
5  docker run -d -p 8000:80 -p 8080:90 -v
   /home/nginx/:/var/log/nginx/ -v
   /home/nginx/config/nginx.conf:/etc/nginx/nginx.con
   f --name nginx  nginx
6
7  ps aux|grep nginx
8
```

##

配置实例_反向代理1

```
1  worker_processes  1;
2  events {
3      worker_connections  1024;
4  }
5
6  http {
7      include       mime.types;
```

```
 8        default_type  application/octet-stream;
 9
10    }
11        sendfile        on;
12        #tcp_nopush        on;
13
14        #keepalive_timeout  0;
15        keepalive_timeout  65;
16
17        #gzip  on;
18
19        server {
20            listen        80;
21            server_name  localhost;
22
23            location / {
24                proxy_pass https://www.baidu.com;
25                expires 3m;
26            }
27
28            #error_page  404              /404.html;
29
30            # redirect server error pages to the
    static page /50x.html
31            #
32            error_page   500 502 503 504  /50x.html;
33            location = /50x.html {
34                root    html;
35            }
36
37
38        }
39
40
41
42 }
```

配置实例_反向代理2

```nginx
worker_processes  1;
events {
    worker_connections  1024;
}


http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    #tcp_nopush     on;
    #keepalive_timeout  0;
    keepalive_timeout  65;
    #gzip  on;
    server {
        listen       80;
        server_name  localhost;

        location / {
            proxy_pass https://blog.csdn.net;
            expires 3m;
        }
        #http://localhost/a/
         location ~/a/ {
             # 设定请求转发的地址 https://www.baidu.com/a
             # 192.68.1.201/a/ ...jgp
            proxy_pass https://www.baidu.com;
        }
        location ~/b/ {
             # 设定请求转发的地址 https://www.cnblogs.com/b
                # 192.68.1.202/b/ ....html.doc
```

```
33              proxy_pass https://www.cnblogs.com;
34          }
35

36          #error_page  404                /404.html;
37

38          # redirect server error pages to the
   static page /50x.html
39          #
40          error_page   500 502 503 504  /50x.html;
41          location = /50x.html {
42              root    html;
43          }
44

45

46      }
47

48
49  }
50
```

配置实例_反向代理3

```
1
2  worker_processes   1;
3  events {
4      worker_connections   1024;
5  }
6

7

8  http {
9      include       mime.types;
10     default_type  application/octet-stream;
11

12

13     sendfile         on;
14     #tcp_nopush       on;
```

```nginx
15
16    #keepalive_timeout  0;
17    keepalive_timeout  65;
18
19    #gzip  on;
20
21    server {
22        listen       80;
23        server_name  localhost;
24
25        location / {
26            proxy_pass https://blog.csdn.net;
27            expires 3m;
28        }
29        #error_page  404              /404.html;
30
31        # redirect server error pages to the
   static page /50x.html
32        #
33        error_page   500 502 503 504  /50x.html;
34        location = /50x.html {
35            root   html;
36        }
37
38
39    }
40 server {
41        listen       90;
42        server_name  localhost;
43
44        location / {
45            proxy_pass https://www.baidu.com;
46        }
47        #error_page  404              /404.html;
48
49        # redirect server error pages to the
   static page /50x.html
```

```
50          #
51          error_page   500 502 503 504  /50x.html;
52          location = /50x.html {
53              root   html;
54          }
55
56
57      }
58
59 }
60
```

配置实例_负载均衡

轮询

```
 1 worker_processes  1;
 2 events {
 3     worker_connections  1024;
 4 }
 5
 6
 7 http {
 8     include       mime.types;
 9     default_type  application/octet-stream;
10
11
12     sendfile        on;
13     #tcp_nopush     on;
14
15     #keepalive_timeout  0;
16     keepalive_timeout  65;
17
18     #gzip  on;
19
20     upstream myserver{
```

```
21            server 192.168.3.201:8080;
22            server 192.168.3.201:8081;
23            }
24      server {
25            listen        80;
26            server_name  localhost;
27

28            location / {
29                proxy_pass http://myserver;
30            }
31

32            #error_page  404              /404.html;
33

34            # redirect server error pages to the
   static page /50x.html
35            #
36            error_page   500 502 503 504  /50x.html;
37            location = /50x.html {
38                root   html;
39            }
40

41

42      }
43

44

45

46  }
```

#启动验证

```
 1   #构建镜像
 2   docker build -t nginx_webapi .
 3   #启动项目
 4   docker run -d -p8080:80 -v
     /home/nginx/Zhaoxi.WebApi/8080/appsettings.json:/
     app/appsettings.json --name
 5   nginx_webapi8080 nginx_webapi
 6   #启动项目
 7   docker run -d -p8081:80 -v
     /home/nginx/Zhaoxi.WebApi/8081/appsettings.json:/
     app/appsettings.json --name nginx_webapi8081
     nginx_webapi
 8
 9   ###http://192.168.3.201:8090/get
10
```

weight权重

可以给每一台服务器设置一个权重，这样权重高的干的活也就会多一点

```
 1   upstream myserver{
 2          server 192.168.3.201:8080 weight=5;
 3          server 192.168.3.201:8081 weight=10;
 4          }
```

ip_hash

这种方式是基于客户端的ip地址，采用hash算法计算下一个请求要选择哪一个服务器，这样固定的ip会访问同一个服务器，可以解决session问题

```
1
2   upstream myserver{
3         ip_hash;
4         server 192.168.3.201:8080;
5         server 192.168.3.201:8081;
6         }
```

least_conn最少链接

会将下一个请求分发到当前链接数最少的一台服务器

```
1   upstream myserver {
2         least_conn;
3         server 192.168.3.201:8080;
4         server 192.168.3.201:8081;
5       }
```

fair

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。

```
1   upstream myserver {
2         fair;
3         server 192.168.3.201:8080;
4         server 192.168.3.201:8081;
5         }
6
```

url_hash

按访问url的hash结果来分配请求，使每个url定向到同一个后端服务器

资源集群服务器,缓存文件,文件存储在第三发服务,则可以缓存到本地服务器

```
1  upstream myserver {
2      hash $request_uri;
3      server squid1:3128;
4      server squid2:3128;
5  }
```

Nginx 配置实例-动静分离

```
1  worker_processes  1;
2  events {
3      worker_connections  1024;
4  }
5
6
7  http {
8      include      mime.types;
9      default_type  application/octet-stream;
10
11
12     sendfile        on;
13     #tcp_nopush     on;
14
15     #keepalive_timeout  0;
16     keepalive_timeout  65;
17
18     #gzip  on;
19
20     server {
21         listen       80;
22         server_name  localhost;
23
24         location / {
25             proxy_pass https://www.baidu.com;
26             expires 3m;
27         }
28          location /www/ {
```

```
            root /data/;
            index index.html index.htm;
        }
         location /image/ {
            root /data/;
            autoindex on;
        }

        #error_page   404                    /404.html;

        # redirect server error pages to the
static page /50x.html
        #
        error_page    500 502 503 504  /50x.html;
        location = /50x.html {
            root    html;
        }



    }



}
```