

Nginx in RDBus

RDBUS中 Nginx的使用

注：使用规则较多，文档中不详细介绍，可参考下方列出的文档链接查看具体使用

1 代理的使用

1.1代理原则

在http节点下，使用upstream配置服务地址，使用server的location配置代理映射。

1.2 使用示例

用8082端口代理172.18.9.33节点的7072端口

```
upstream rest {
    server 172.18.9.33:7072;
}
server {
    server_name      kafka.rest.proxy ;
    listen 8082;
    location / {
        proxy_pass      http://rest;
        proxy_set_header Host    $http_host;
    }
}
```

1.3 参考文档

<https://www.cnblogs.com/fanzhidongyzby/p/5194895.html>

2负载均衡

2.1策略

策略	方式	功能
轮询	默认方式	每个请求会按时间顺序逐一分配到不同的后端服务器
weight	权重方式	在轮询策略的基础上指定轮询的几率
ip_hash	依据ip分配方式	指定负载均衡器按照基于客户端IP的分配，保证每个访客都固定访问一个后端服务器，解决session不能跨服务器的问题
least_conn	最少连接方式	把请求转发给连接数较少的后端服务器
fair（第三方）	响应时间方式	按照服务器端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）	依据URL分配方式	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，要配合缓存命中来使用

2.2使用示例

用8082端口负载均衡 172.18.9.33-35节点的7072端口

```
upstream rest {
    server 172.18.9.33:7072;
    server 172.18.9.34:7072;
    server 172.18.9.35:7072;
}
server {
    server_name      kafka.rest.proxy ;
    listen 8082;
    location / {
        proxy_pass      http://rest;
        proxy_set_header    Host      $http_host;
    }
}
```

2.3 参考文档

<https://www.cnblogs.com/1214804270hacker/p/9325150.html>

3路由重写

3.1原则

server节点下，操作location节点，使用nginx的重写规则进行重写（）

3.2使用示例

将/BASIC/abcd重写为/abcd

方式一

```
server {
    location /BASIC/ {
        rewrite ^/BASIC/?(.*)$ /$1 break;
    }
}
```

方式二

```
server {
    location /BASIC/ {
        rewrite ^.+BASIC/?(.*)$ /$1 break;
    }
}
```

3.3参考文档

<https://www.cnblogs.com/dalaoyang/p/10264919.html>

<https://www.cnblogs.com/sunmmi/articles/7217050.html>

4 日志使用

4.1 日志分类

4.1.1 访问日志

作用：记录客户端的请求

语法：access_log path [format [buffer=size] [gzip[=level]] [flush=time] [if=condition]]; # 设置访问日志

access_log off; # 关闭访问日志

作用域：http, server, location, limit_except

4.1.2 错误日志

作用：记录服务器和请求处理过程中的错误信息

语法：error_log path(存放路径) level(日志等级)

日志等级：debug, info, notice, warn, error, crit, alert, emerg

作用域：main, http, mail, stream, server, location

4.2 自定义日志格式示例

格式

```
log_format test '[$time_iso8601] $remote_addr "$http_referer" "$request" ';
```

```
access_log logs/nginx-access.log test;
```

日志输出

```
[2021-01-12T12:16:10+08:00] 66.16.41.109 "http://66.16.41.109:8080/" "GET /dashboard HTTP/1.1"
```

4.3 参考文档

<https://blog.51cto.com/longlei/2132170>

https://blog.csdn.net/weixin_39841640/article/details/111807657

5 访问控制

5.1 referer

作用：限制访问来源

示例：只允许66.16.41.109访问，不允许浏览器直接输入url访问

```
location / {
    valid_referers 66.16.41.109 ;
    if ($invalid_referer) {
        return 403;
    }
    index index.html;
}
```

bug：当有nginx缓存时，此方式可能会无法禁止缓存过的页面

解决：禁止浏览器缓存

```
location / {
    add_header Cache-Control no-cache;
    valid_referers 66.16.41.109 ;
    if ($invalid_referer) {
        return 403;
    }
    index index.html;
}
```

5.2黑白名单

作用：允许白名单ip访问，禁止黑名单ip访问

示例：只允许192.168.1.0/24、10.1.1.0/16网段的主机访问，拒绝其他所有

```
location / {
    deny 192.168.1.0/24;    # 黑名单
    deny 10.1.1.0/16;      # 黑名单
    allow all;             # 白名单
}
```

5.3参考文档

<https://blog.csdn.net/zhuyu19911016520/article/details/91352154>

<https://www.cnblogs.com/kaishirensi/p/11132975.html>

- RDBUS中 Nginx的使用
 - 1 代理的使用
 - 1.1代理原则
 - 1.2 使用示例
 - 1.3 参考文档
 - 2负载均衡
 - 2.1策略
 - 2.2使用示例
 - 2.3 参考文档
 - 3路由重写
 - 3.1原则
 - 3.2使用示例
 - 3.3参考文档
 - 4日志使用
 - 4.1日志分类
 - 4.1.1
访问日志
 - 4.1.2
错误日志
 - 4.2自定义日志格式示例
 - 5访问控制
 - 5.1referer
 - 5.2黑白名单
 - 5.3参考文档