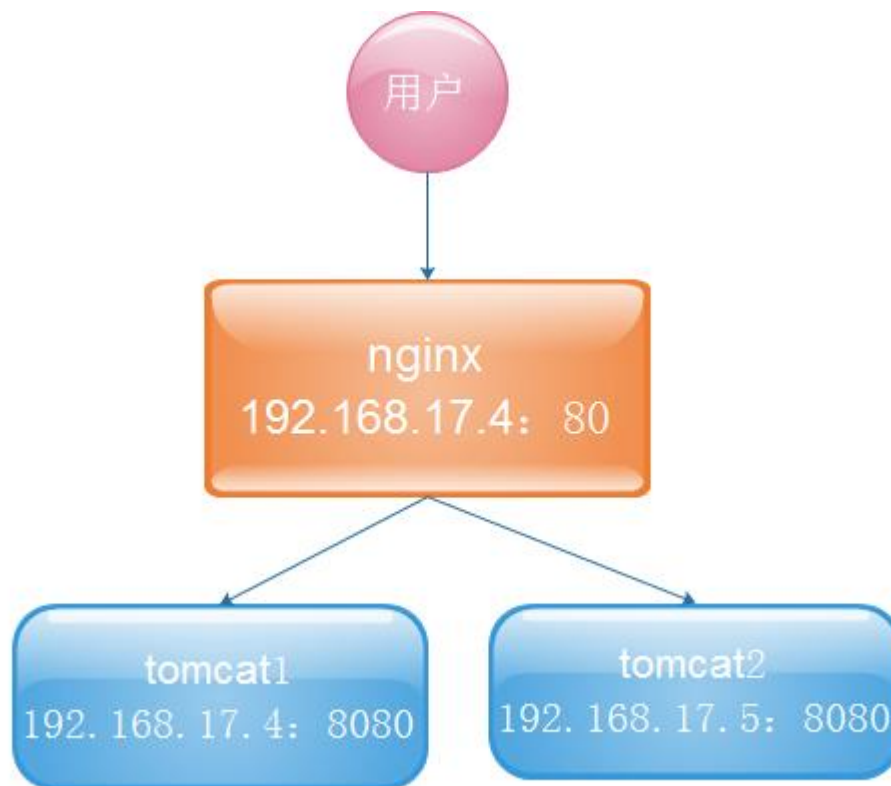


Nginx和高并发

加入尚学堂，一起进步！

- 经典的反向代理结构



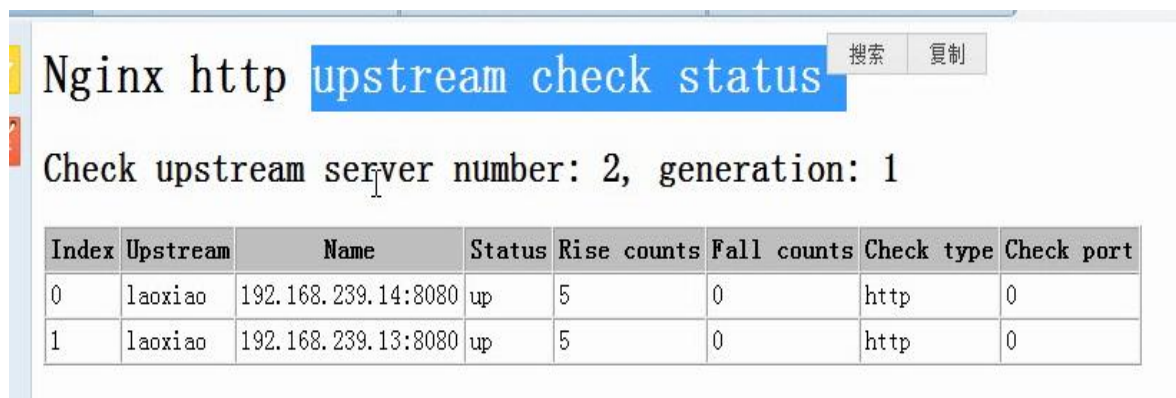
- 虚拟主机
- 一个server{} 就是一个虚拟主机
- 基于域名的

- 反向代理：
- proxy_pass

- 反向代理配置nginx.conf :
 - upstream 名字 {
 - server IP:PORT;
 - server IP:PORT;
 - }
 - server {
 - location / {
 - proxy_pass http://名字;
 -
 - }
 - }

- `tengine`新增健康检查模块
- 配置一个status的location
- `location /status {`
- `check_status;`
- `}`
- 在upstream配置如下
- `check interval=3000 rise=2 fall=5 timeout=1000 type=http;`
- `check_http_send "HEAD / HTTP/1.0\r\n\r\n";`
- `check_http_expect_alive http_2xx http_3xx;`
-

- 对后端服务器健康检查



The screenshot shows the output of the 'Nginx http upstream check status' command. It includes a search bar with '搜索' (Search) and '复制' (Copy) buttons. Below the command, it states 'Check upstream server number: 2, generation: 1'. A table follows, displaying the status of two upstream servers. The table has columns for Index, Upstream, Name, Status, Rise counts, Fall counts, Check type, and Check port.

Index	Upstream	Name	Status	Rise counts	Fall counts	Check type	Check port
0	laoxiao	192.168.239.14:8080	up	5	0	http	0
1	laoxiao	192.168.239.13:8080	up	5	0	http	0

- 对后端服务器健康检查的配置

```
http {  
    upstream cluster1 {  
        # simple round-robin  
        server 192.168.0.1:80;  
        server 192.168.0.2:80;  
  
        check interval=3000 rise=2 fall=5 timeout=1000 type=http;  
        check_http_send "HEAD / HTTP/1.0\r\n\r\n";  
        check_http_expect_alive http_2xx http_3xx;  
    }  
}
```

```
location /status {  
    check_status;  
  
    access_log off;  
    allow SOME.IP.ADD.RESS;  
    deny all;  
}
```


- 对后端服务器健康检查的配置
 - upstream cluster2
 - {
 - # simple round-robin
 - server 192.168.0.3:80;
 - server 192.168.0.4:80;
 - check interval=3000 rise=2 fall=5 timeout=1000 type=http;
 - check_keepalive_requests 100;
 - check_http_send "HEAD / HTTP/1.1\r\nConnection: keep-alive\r\n\r\n";
 - check_http_expect_alive http_2xx http_3xx;
 - }
 - server {
 - listen 80;
 - location /status {
 - check_status;
 - }
 - }