

## 一、方式1：轮询

轮询是 **nginx** 负载默认的方式，它的本意是将海量请求按照时间的顺序分配到不同的服务器上；如果某个服务器down掉，nginx可以将其自动踢出集群。

```
#gzip on;
upstream study {
    server 127.0.0.1:8080 ;
    server 127.0.0.1:8081 ;

    server
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    html;
        proxy_pass http://study;          #代理，让请求进入s
        index   index.html index.htm;    https://blog.csdn.net/Kevinnsn
    }
}
```

访问时，nginx会自动轮询这两个服务

由此便实现了负载均衡

## 二、方式2：权重

指定每个服务器的权重比例，由于有的服务器性能比较好，所以通过设置不同服务器间权重的比例可以发挥服务器的最大性能。

```
keepalive_timeout 65;

#gzip on;
upstream study {
    server 127.0.0.1:8080 weight = 1;
    server 127.0.0.1:8081 weight = 2;
}
#
server
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;
    #access_log logs/host.access.log main;

    location / {
        root    html;
        proxy_pass http://study;https://blog.csdn.net/Kevinnsn
        index   index.html index.htm;
    }
}
```

指定权重比例

这样8080服务就会有1/3的权重比例，8081服务就会有2/3的权重比例（当8080计数到1，如果有请求就会去找8081服务）

### 方式3: iphash

根据ip经过 **hash** 后的结果进行分配, 这样每个ip只能固定访问一个服务

```
#gzip on;
upstream study {
    ip_hash;
    server 127.0.0.1:8080 ;
    server 127.0.0.1:8081 ;

    server
server {
    listen      80;
    server_name localhost;
    https://blog.csdn.net/Kevinnsn
```

← → ↻ ⚠ 不安全 | 192.168.160.128

应用 csdn 专业it技术社... MVN MavenRepository Spr

8081

一直刷新, 只能访问8081服务

https://blog.csdn.net/Kevinnsn

### 方式4: 最小连接

将请求分配到连接最少的服务上

```
keepalive_timeout 65;

#gzip on;
upstream study {
    least_conn;
    server 127.0.0.1:8080 ;
    server 127.0.0.1:8081 ;
}

# server
server {
```

← → ↻ ⚠ 不安全 | 192.168.160.128

应用 csdn 专业it技术社... MVN MavenRepository

8081

← → ↻ ⚠ 不安全 | 192.168.160.128

应用 csdn 专业it技术社... MVN MavenRepository

8080

其结果类似于轮询