

## 20 负载均衡模块搭建

更新时间：2020-02-18 10:21:54



“

学习这件事不在乎有没有人教你，最重要的是在于你自己有没有觉悟和恒心。——法布尔

”

### 前言

上一篇文章中我们介绍了 **负载均衡** 出现的原因，作用已经常见的负载均衡策略。这一篇文章我们将详细的介绍一下 **Nginx** 中实现负载均衡机制的模块。

### 负载均衡模块

在 `Nginx` 中，通过 `ngx_http_upstream_module` 实现负载均衡功能。这个模块的指令很少，对应的是我们在上篇文章中讲到的各个概念。

## Example Configuration

### Directives

[upstream](#)

[server](#)

[zone](#)

[state](#)

[hash](#)

[ip\\_hash](#)

[keepalive](#)

[keepalive requests](#)

[keepalive timeout](#)

[ntlm](#)

[least\\_conn](#)

[least time](#)

[queue](#)

[random](#)

[resolver](#)

[resolver timeout](#)

[sticky](#)

[sticky\\_cookie insert](#)

### Embedded Variables

我们在本文中重点介绍其中几个常用指令。

### 后端集群指令

`upstream` 和 `server` 指令互相配合，用于配置一个我们在上篇文章中提到的后端集群。这个集群中的所有服务器都提供相同的功能，`Nginx` 可以通过不同的负载均衡策略将请求分发给不同的后端服务器。因为针对不同的集群，我们可以使用不同的负载均衡策略，所以负载均衡策略是和 `upstream` 配合使用的。

`server` 指令

`server` 指令用于指定集群中的后端服务器实例。我们可以使用域名，`ip` 及端口的方式进行配置。服务器还有很多属性，我们可以通过这些属性字段对服务器进行修饰，比如权重，最大连接数，失败连接数等等。



我们比较常用的 `权重` 这个属性。我们在实际的项目中，有些机器可能性能很好，可以支持很多的请求。而有些旧机器可能无法支持太多的请求，这个时候我们就可以通过 `权重weight` 来对每个机器分配不同的权重。

其实这里面的指令都是和实际中的概念一一对应的

```
upstream backend {  
    # 下面定义了四个后端服务器，形成了一个后端集群  
    server backend1.example.com weight=5;  
    server backend2.example.com max_fails=3;  
    server backend3.example.com;  
    server 127.0.0.1: 8080; #ip+端口的格式  
}
```

## 负载均衡策略

`Nginx` 中支持的几种负载均衡策略，其中 `轮询` 策略是默认的负载均衡策略，我们在定义后端集群的时候可以不用写（上面的示例就是一个使用了默认的 `轮询` 策略）。



对于上面的例子来说，`Nginx` 将使用加权轮训的机制在所有后端服务器之间分发请求。每 `8` 个请求的分发如下：

- `5` 个请求被分发给 `backend1.example.com`
- 剩下的 `3` 个请求被平均分配给了其余三个后端服务器
- 当一个后端服务器发生错误的时候，会选择下一个服务器，知道所有的服务器都被重试。

## 总结

我们在这一篇文章中介绍了 **Nginx** 中实现负载均衡相关功能的模块以及控制指令。下一篇文章中我们将结合实战来详解的理解相关的概念和这些指令。

}