

## Nginx课程笔记 (讲师: 应癡)

### 主要课程内容

- 第一部分: `kgw` 基函数 `kgw` 是什么? 能做什么事情? 返回什么头信息? 等等
  - 第二部分: `kgw000` 与 `kgw` 关系讲解
  - 第三部分: `kgw` 基函数之 `kgw000`
  - 第四部分: `kgw` 基函数之 `kgw000`
  - 第五部分: `kgw` 基函数之 `kgw000`
  - 第六部分: `kgw` 基函数之 `kgw000`
- `kgw` 函数与 `kgw000` 函数关系图, 所以我们要了解 `kgw` 函数与 `kgw000` 函数的关系了。

## 第一部分 Nginx基础回顾

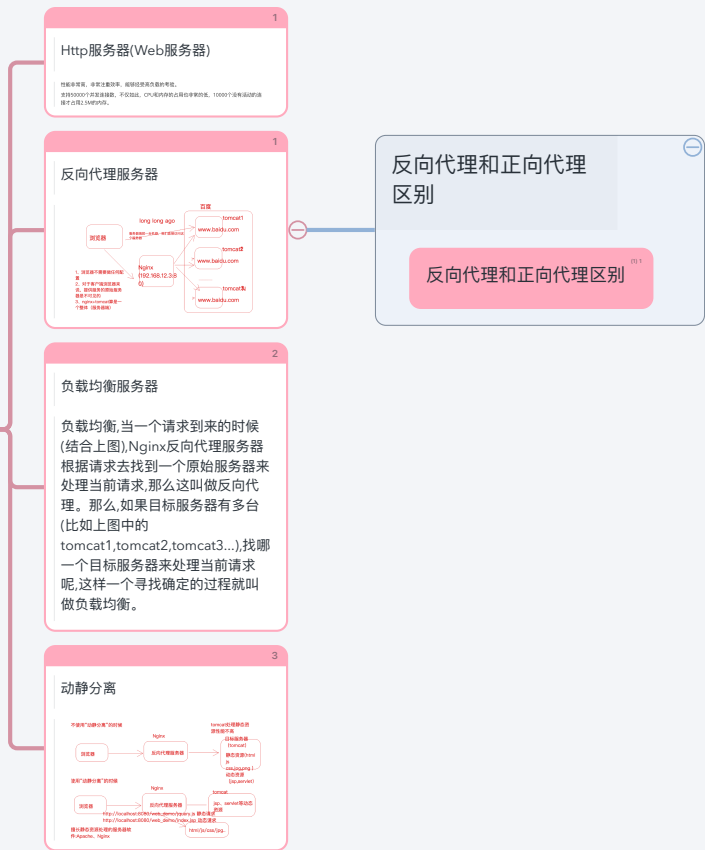
## Nginx 到底是什么？

Nginx 是一个高性能的HTTP和反向代理web服务器,核心特点是占有内存少,并发能力强

## Nginx的安装

## Nginx主要命令

- `nginx` 跟 `nginx`
- `nginx -s stop` 停止 `nginx` (当然也可以找到 `nginx` 进程号, 然后使用 `kill -s stop nginx` 进程)
- `nginx -s reload` (重新加载 `nginx.conf` 配置文件)



## Nginx 的特点

- ◆ 跨平台: Nginx可以在大多数类unix操作系统上编译运行, 而且也有windows版本
- ◆ Nginx的上手和配置简单, 配置也比较简单
- ◆ 高并发, 性能好
- ◆ 稳定的内核结构, 低资源消耗

## 第二部分 Nginx核心 配置文件解读

全局块

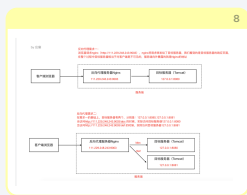
events块



http块



### 第三部分 Nginx应用场景之反向代理



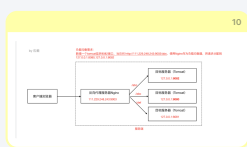
location 语法如下:

在R语言官方文档中，location主要有以下几种形式：

- 1) 正则匹配 `location ~ Agouo { }`
- 2) 不分大小写的正则匹配 `location ~* Agouo { }`
- 3) 匹配连续的位置 `location ~+ Agouo { }`
- 4) 精确匹配 `location = Agouo { }`
- 5) 普通路径匹配正则 `location location Agouo { }`

• 优缺点

## 第四部分 Nginx应用 场景之负载均衡



## Nginx负载均衡策略

轮询

```
upstream lagouServer{
    server 111.229.248.243:8080;
    server 111.229.248.243:8082;
}

location /abc {
    proxy_pass http://lagouServer/;
}
```

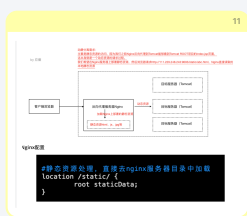
weight

```
upstream lagosServer{
    server 111.229.248.243:8080 weight=1;
    server 111.229.248.243:8082 weight=2;
}
```

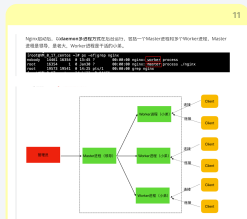
## ip\_hash

```
upstream lagouServer{
    ip_hash;
    server 111.229.248.243:8080;
    server 111.229.248.243:8082;
}
```

## 第五部分 Nginx应用 场景之动静分离



## 第六部分 Nginx底层 进程机制剖析



master进程

- 修改外部信号为低电平，worker进程发送信号 (signal = 0) 给os
- 当os收到信号时，将worker进程设置为等待状态 (wait) 直到收到信号 (signal = 1) 为止

worker进程

- worker 进程彼此间有需求。多个 worker 进程之间是协作的，它们同时竞争需求，各自处理这些需求的一部分。一个需求，只能由一个 worker 进程来处理，不可能同时被多个 worker 进程处理。worker 进程的个数是可以调整的，一般设置与机器

以 `./nginx -s reload` 来说  
明nginx信号处理这部分

- (1) docker安装与配置主节点环境
- (2) 学习原理：包括学习了容错机制、驱动方式(集群的选举机制)
- (3) 修改主机网络参数配置，给docker设置桥接网
- (4) 搭建成功，使用docker安装并运行redis
- (5) 介绍redis特有的命令及数据类型，我通过自己搭建的测试环境来使用redis的cluster功能进行了验证。

## worker进程处理请求部分的说明

此时，测试端与网关接口，一个请求过来时，请求传多了worker进程，那么多了worker进程可接收这个请求。

- `nginx`的配置文件则保证只有一个`worker`进程能够处理请求，拿到该请求的那个进程负责处理该请求，而该进程使用`accept`接收该请求，然后解码、处理，返回响应给客户端。