3W1H - 学习的万能公式

解决什么问题？

1. What - 负载均衡器 - 分发用户请求

2. Where/Who/When - 当有多台服务器提供的服务的时候（什么场景下面使用）

3. Why - Nginx/LVS/F5/HA/DNS/

第七层 -- http处理，特性多，nginx 发展最迅猛~~

4. How - 底层实现原理

1. 响应 header头部信息加上 服务器的唯一编号，排查问题有帮助
2. Nginx配置 ---

Work关系到资源利用 – 实际处理请求的进程

多进程 -- 与CPU的绑定 – 提高nginx运作效率  
worker\_processes auto;

worker\_cpu\_affinity auto;

最大连接数– max\_connections 3000 --- 实际我们的压力大约控制在 3000qps

1. Nginx角色 --- 除了分流之外，往往还会承担其他事情

css合并/   
LUA + Nginx – 功能增强 – 比较常见 openresty

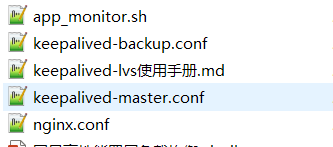
|  |
| --- |
| lua\_package\_path "/usr/local/lib/lua/?.lua;;/usr/local/nginx-1.14.2/lua\_scripts/?.lua;;";  server {  listen 80;  server\_name localhost;  #charset koi8-r;  #access\_log logs/host.access.log main;  location /lua {  content\_by\_lua\_block {  local mydata = require "mydata"  ngx.say(mydata.get\_age("dog"))  }  }  location /get-msg {  content\_by\_lua\_block {  local request\_method = ngx.var.request\_method  local args = nil  local key = nil  local value = nil  if "GET" == request\_method then  args = ngx.req.get\_uri\_args()  elseif "POST" == request\_method then  ngx.req.read\_body()  args = ngx.req.get\_post\_args()  end  key = args["key"]  local redis = require "resty.redis"  local red = redis:new()  red:set\_timeout(1000) -- 1 sec  local ok, err = red:connect("127.0.0.1", 6379)  if not ok then  ngx.say("failed to connect: ", err)  return  end  local res, err = red:get(key)  if not res then  ngx.say("failed to get "..key..": ", err)  return  end  if res == ngx.null then  ngx.say(key.." not found.")  return  end  ngx.say("get from redis >>> "..key..": ", res)  local ok, err = red:close()  if not ok then  ngx.say("failed to close:", err)  return  end  }  }  location / {  proxy\_set\_header Host $http\_host;  proxy\_pass http://app\_servers;  } |

Lua…第二语言  
层次 – 快速理解，java概念去理解，偶尔接触一下  
系统性学习 – 语法…库…

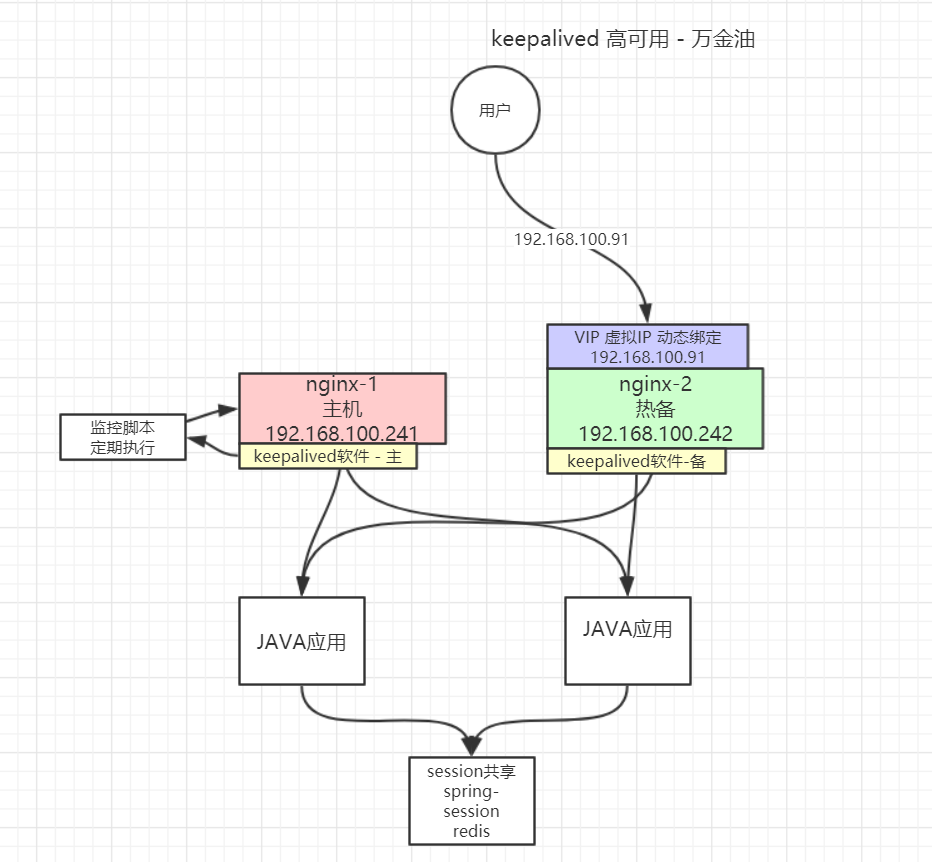
1. 网易云的应用  
   硬件层面，操作系统层面+负载均衡软件层面（lvs，dpdk）  
   F5，思科，华为，浪潮….网络设备 – 防火墙，负载均衡的功能~~ （路由器 – 实现网络数据包分发~~CPU处理器和普通的I7，因特尔至强~~区别）

Nginx – 解析http -- 意味消耗的资源比较多 --- 内存 CPU

1. keepalived 高可用 - 万金油

高可用 / 负载均衡 区分  
keepalived 争抢vip + 检测程序是否可用  
 . 主备之分  
 . 启动 需要配置  
 

目的：监控某种软件的健康情况，如果不正常，则切换

~ 双机热备 ~ 多机热备 ~   
  
注意网卡接口名称