**keepalived热备**

keepalived概述

keepalived概述

调度器出现单点故障，如何解决？

keepalived实现了高可用集群

keepaliverd最初是为LVS设计的，专门监控各服务器节点的状态

keepalived后来加入了VRRP功能，防止单点故障

keepalived运行原理

keepalived检测每个服务器节点状态

服务器节点异常或工作出现故障，keepalived将故障节点从集群系统中剔除

故障节点恢复后，keepalived再将其加入到集群系统中

所有工作自动未完成，无需人工干预

keepalived服务

keepalived安装

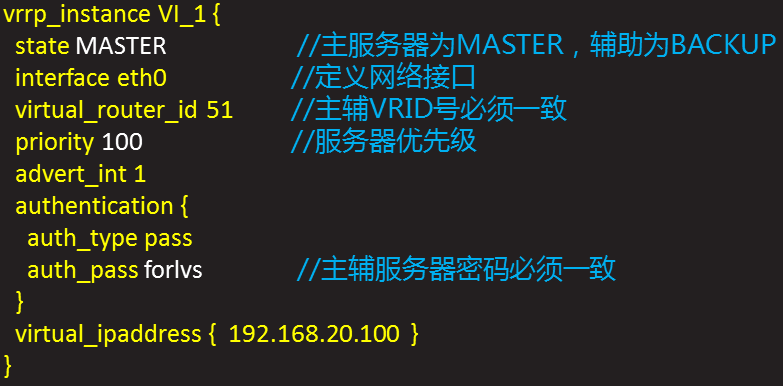
rhel7的光盘中已经包含keepalived软件包，只要配置好yum，指向光盘源即可安装

yum -y install keepalived

配置文件解析

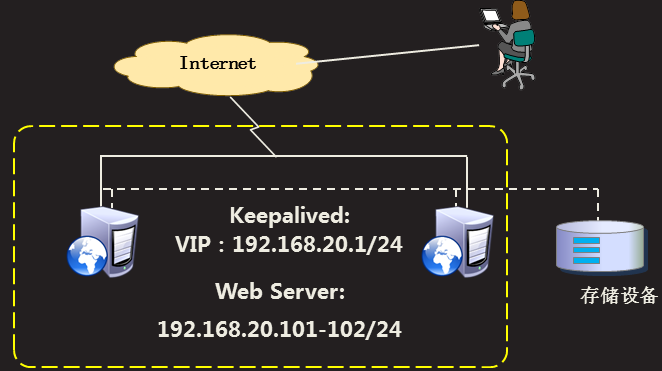
/etc/keepalived/keepalived.conf





高可用web拓扑

使用keepalived为主从社而被提供VIP地址漂移



案例

配置web服务器

（web1）ifconfig eth0 192.168.20.101

yum -y install httpd

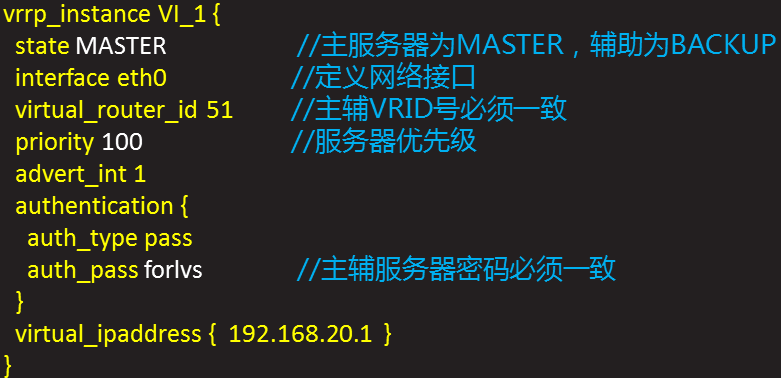
systemctl start httpd;systemctl enable httpd

（web2）ifconfig eth0 192.168.20.102

yum -y install httpd

systemctl start httpd;systemctl enable httpd

使用keepalived为服务器提供VIP



**keepalived+lvs**

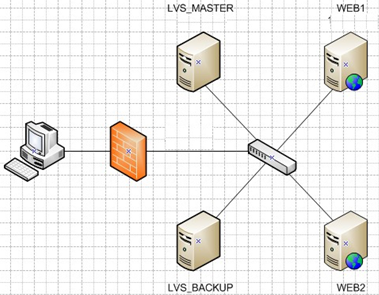
高可用调度器

keepalived+lvs拓扑

使用keepalived高可用解决调度器单点失败问题

主、备调度器上配置LVS

主调度器异常时，keepalived启用备用调度器



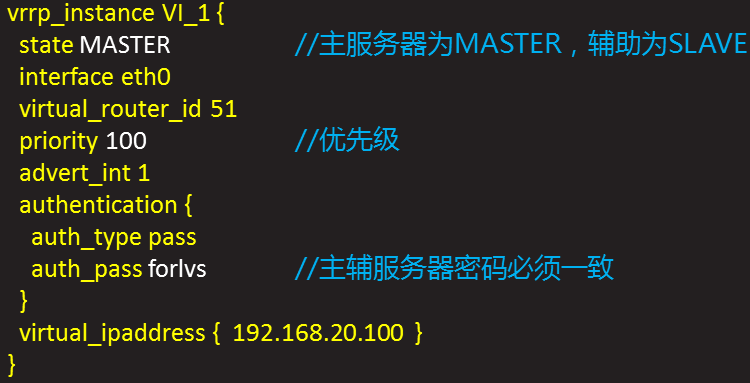
keepalived配置说明

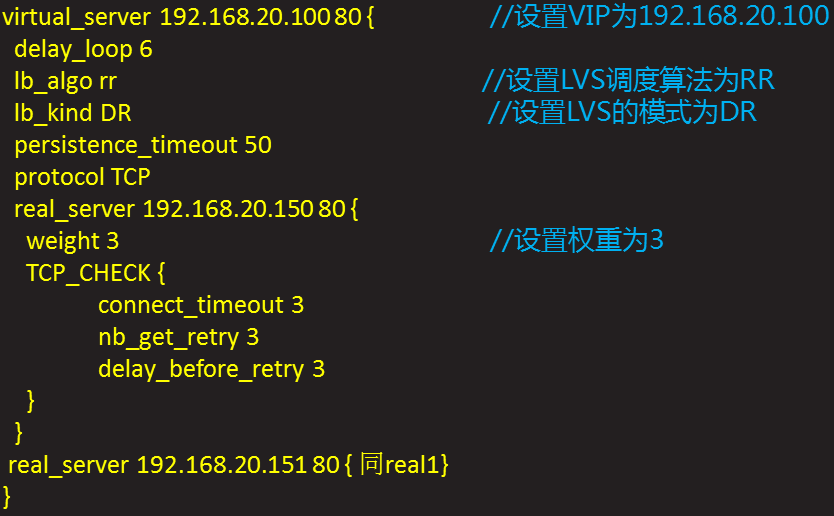
LVS相关信息通过keepalived配置即可

主要配置文件说明如下



VRRP实例设置





real server配置

真实服务器运行在DR模式下

修改内核参数，并附加VIP

详细配置参见LVS相关章节

**HAProxy服务器**

HAProxy概述

HAProxy简介

它是免费、快速并且可靠的一种解决方案

适当与那些负载特大的web站点，这些站点通常又需要会话保持或七层处理

提高高可用、负载均衡以及TCP和HTTP应用的代理

衡量负载均衡器性能的因素

session rate会话率

每秒钟产生的会话数

session concurrency并发会话数

服务器处理会话的时间越长，并发会话数越多

DaTa rate数据速率

以MB/s或Mbps衡量

大的对象导致并发会话数增加

高会话数、高数据速率要求更多的内存

HAProxy工作模式

mode http

客户端请求被深度分析后再发往服务器

mode tcp

客户端与服务器之间建立会话，不检查第七层信息

mode health

仅做检查状态检查，已经不建议使用

HTTP协议解析

HTTP解析

当HAProxy运行在HTTP模式下，HTTP请求（request）和响应（response）均被完全分析和索引，这样便于创建恰当的匹配规则

理解HTTP请求和响应，对于更好的创建匹配规则至关重要

HTTP事务模型

HTTP协议是事务驱动的

每个请求仅能对应一个响应

常见模型

HTTP close

客户端向服务器建立一个tcp连接

客户端发送请求给服务器

服务器响应客户端请求后即断开连接

如果客户端到服务器的请求不只一个，那么就要不断的去建立连接

TCP三次握手消耗相对较大的系统资源，同时延迟较大

Keep-alive

一次连接可以传输多个请求

客户端需要知道传输内容的长度，以避免无限期的等待传输结束

降低两个HTTP事务间的延迟

需要相对较少的服务器资源

Pipelining

仍然使用Keep-alive

在发送后续请求前，不用等前面的请求已经得到回应

适用于有大量图片的页面

降低了多次请求之间的网络延迟

HTTP头部信息

请求头部信息

方法：GET

URI：/serv/login.php?lang=en&profile=2

版本：HTTP/1.1

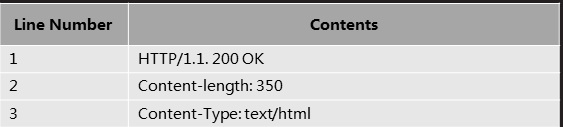


响应头部信息

版本：HTTP:HTTP/1.1

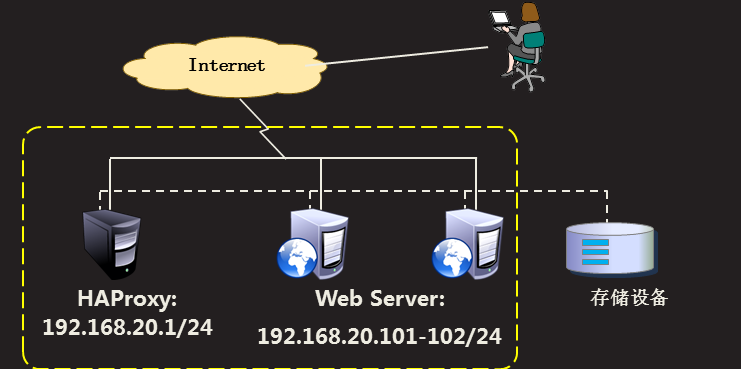
状态码：200

原因：ok



HAProxy配置实例

项目拓扑图



HAProxy安装

yum install haproxy

配置文件说明

HAProxy配置参数来源

命令行：总是具有最高优先级

global部分：全局设置进程级别参数

代理声明部分

来自于default、listen、frontend和backend

配置文件可由如下部分构成

defaults

为后续的其他部分设置缺省参数

缺省参数可以被后续部分重置

frontend

描述接受客户端侦听套接字（socket）集

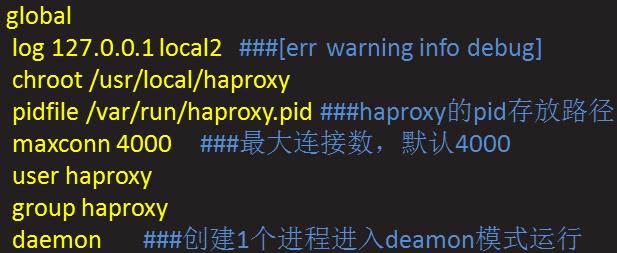
backend

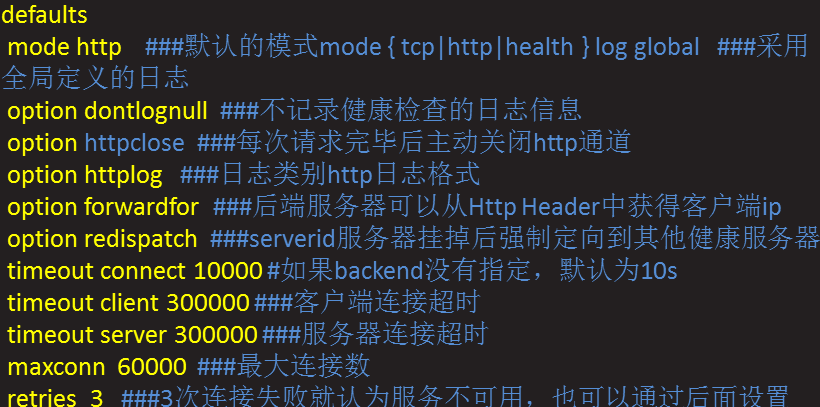
描述转发链接的服务器集

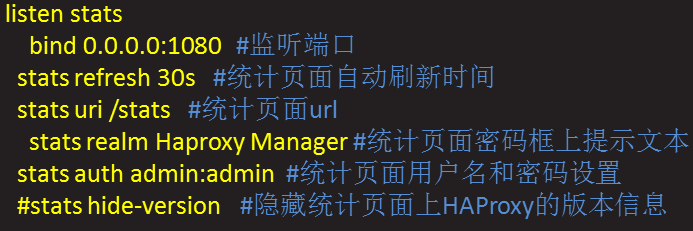
listen

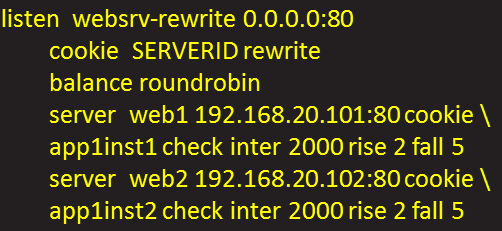
把frontend和backend结合到一起的完整声明

/etc/haproxy/haproxy.cfg









管理服务

启动服务

systemctl start haproxy

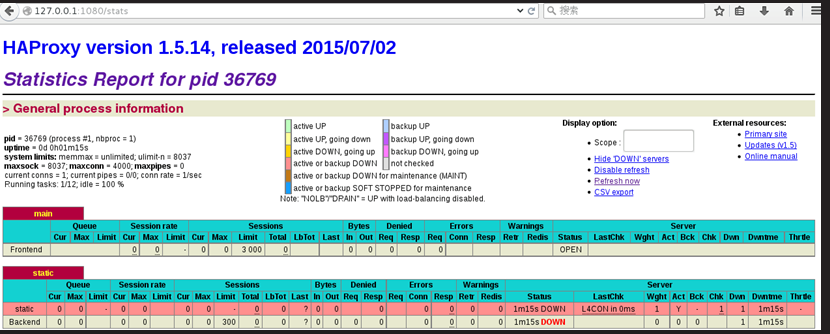
停止服务

systemctl stop haproxy

查看状态

systemctl status haproxy

监控HAProxy状态



集群调度软件对比

nginx分析

优点：

工作在7层，可以针对http做分流策略

正则表达式比HAProxy强大

安装、配置、测试简单，通过日志可以解决多数问题

并发量可以达到几万次

nginx还可以作为web服务器使用

缺点：

仅支持http、httpsd、mail协议，应用面小

监控检查仅通过端口，无法使用url检查

LVS分析

优点

负载能力强，工作在4层，对内存、CPU消耗低

配置性低，没有太多可配置性，减少人为错误

应用面广，几乎可以为所有应用提供负载均衡

缺点

不支持正则表达式，不能实现动静分离

如果网站架构庞大，LVS-DR比较繁琐

HAProxy分析

优点

支持session、cookie功能

可以通过url进行健康检查

效率、负载均衡速度，高于nginx，低于LVS

HAProxy支持TCP，可以对mysql进行负载均衡

调度算法丰富

缺点

正则弱于nginx

日志依赖于syslogd，不支持apache日志