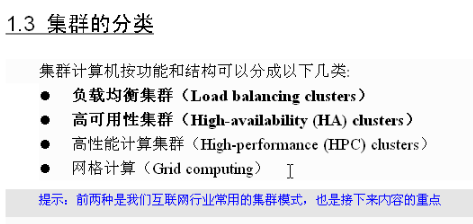
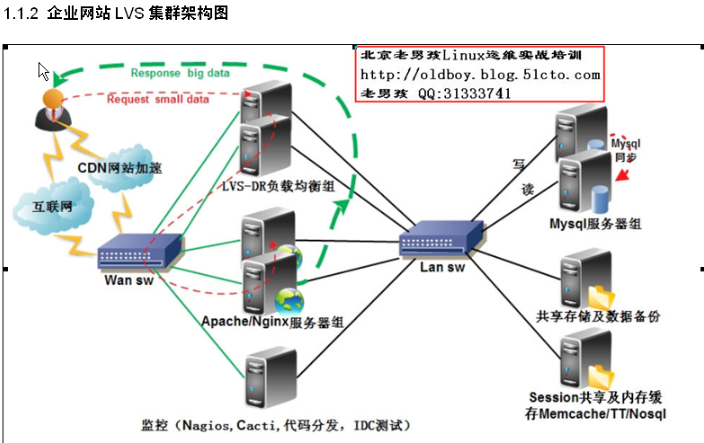
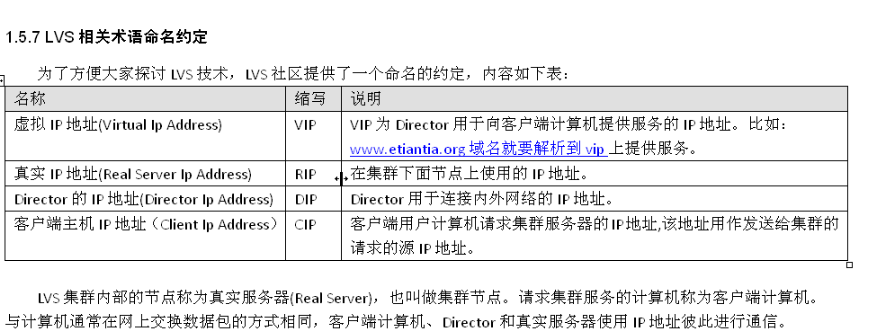
# LVS负载均衡

## 原理







LVS负载均衡集群的3种工作模式：

1. NAT模式-网络地址转换（VS/NAT）:

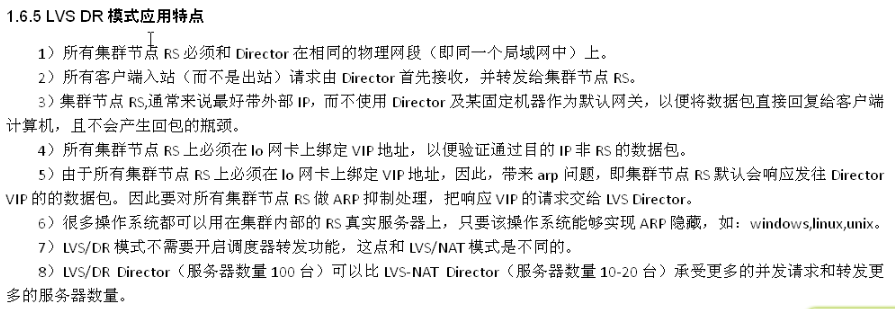
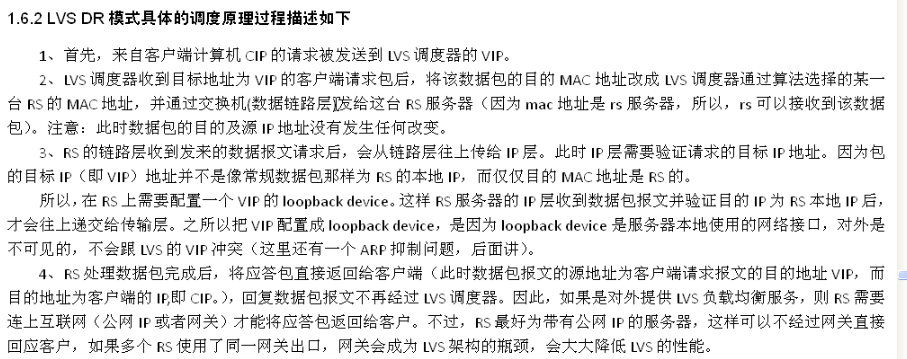
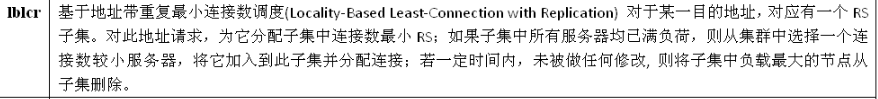
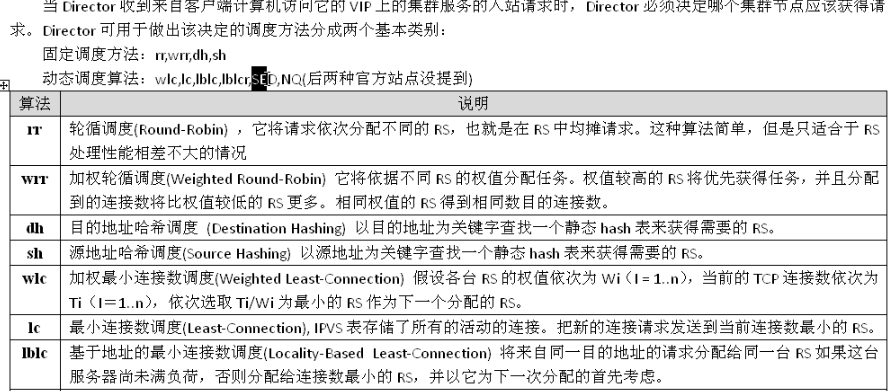
将一组服务器构成一个高性能、高可用的虚拟服务器，类似公路上的收费站，来去都要经过LB负载均衡器，通过修改目的地址、端口或源地址或源端口，一般处理10-20台服务器。

1. （VS/TUN）

通过IP隧道实现虚拟服务器的方法，因为才NAT技术时，由于请求和响应报文都必须经过LB，当客户请求越来越多时，LB的处理能力将成为瓶颈。为了解决这个问题，LB将请求报文通过IP隧道转发至真实服务器，而真实服务器将响应直接返回给客户端，这样LB只处理请求报文。由于一般网络应答数据比请求报文大很多，采用TUN后，集群系统的最大吞吐量可提高10倍。

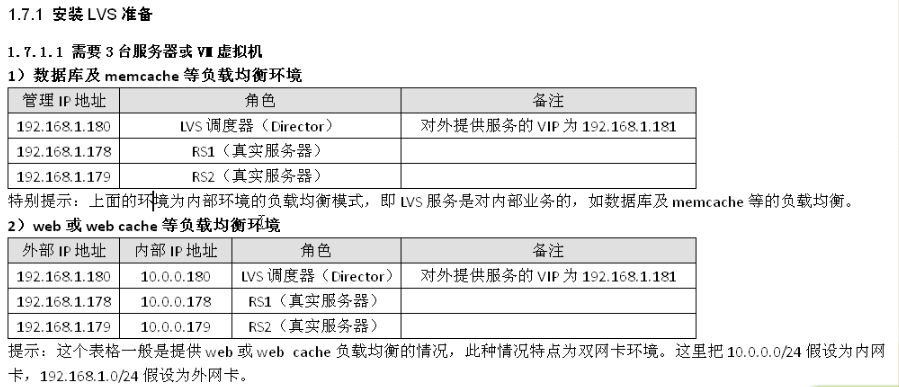
1. （VS/DR）

通过直接路由器实现虚拟服务器。VS/DR通过改写请求报文的MAC地址，将请求发送到真实服务器，而真实服务器将响应直接发给客户端。同VS/TUN一样，VS/DR技术可以极大的提高集群系统的伸缩性，这种方法没有IP隧道的开销，对集群的真实服务器也没有必须支持IP隧道协议的要求，但是要求LB与真实服务器都有一块网卡连在同一物理段上。

LVS调度算法决定了如何在这些集群节点之间分布工作负荷。

## 安装配置

### 现在LB服务器上安装IPVS



先下载keepalived和ipvsadm-1.24文件

1. 先查看系统内核



1. 然后查看文件



1. 对内核进行软链接

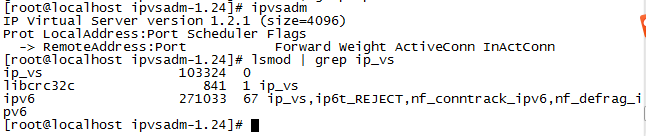


1. 进入ipvsadm-1.24目录下进行make && make install
2. 此时用lsmod 查看是否有加载ip\_vs模块



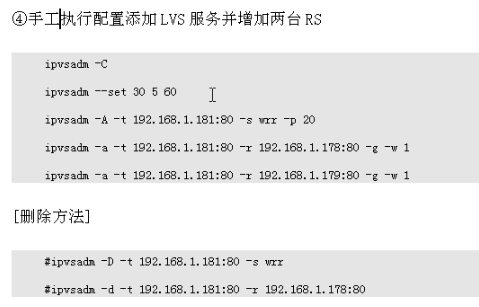
显示为空，表示还未加载。

1. 执行ipvsadm命令把ip\_vs加载到系统内核，或者执行modprobe ip\_vs也可以加载，加载后再执行lsmod就可以看到已经被加载了。出现这个就表示已经安装好了





两台RS服务器和LB服务器L0地址要完全一样。



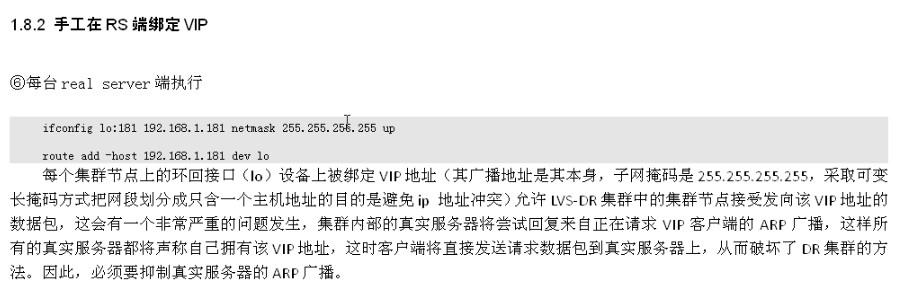
参数说明：

-A --add-service 在内核的虚拟服务器表中添加一条新的虚拟服务器记录。也  
就是增加一台新的虚拟服务器。  
-E --edit-service 编辑内核虚拟服务器表中的一条虚拟服务器记录。  
-D --delete-service 删除内核虚拟服务器表中的一条虚拟服务器记录。  
-C --clear 清除内核虚拟服务器表中的所有记录。  
-R --restore 恢复虚拟服务器规则  
-S --save 保存虚拟服务器规则，输出为-R 选项可读的格式  
-a --add-server 在内核虚拟服务器表的一条记录里添加一条新的真实服务器  
记录。也就是在一个虚拟服务器中增加一台新的真实服务器  
-e --edit-server 编辑一条虚拟服务器记录中的某条真实服务器记录  
-d --delete-server 删除一条虚拟服务器记录中的某条真实服务器记录  
-L|-l --list 显示内核虚拟服务器表  
-Z --zero 虚拟服务表计数器清零（清空当前的连接数量等）  
--set tcp tcpfin udp 设置连接超时值  
--start-daemon 启动同步守护进程。他后面可以是master 或backup，用来说  
明LVS Router 是master 或是backup。在这个功能上也可以采用keepalived 的  
VRRP 功能。  
--stop-daemon 停止同步守护进程  
-h --help 显示帮助信息  
其他的选项:  
-t --tcp-service service-address 说明虚拟服务器提供的是tcp 的服务  
[vip:port] or [real-server-ip:port]  
-u --udp-service service-address 说明虚拟服务器提供的是udp 的服务  
[vip:port] or [real-server-ip:port]  
-f --fwmark-service fwmark 说明是经过iptables 标记过的服务类型。  
-s --scheduler scheduler 使用的调度算法，有这样几个选项  
rr|wrr|lc|wlc|lblc|lblcr|dh|sh|sed|nq,  
默认的调度算法是： wlc.  
-p --persistent [timeout] 持久稳固的服务。这个选项的意思是来自同一个客  
户的多次请求，将被同一台真实的服务器处理。timeout 的默认值为300 秒。  
-M --netmask netmask persistent granularity mask  
-r --real-server server-address 真实的服务器[Real-Server:port]  
-g --gatewaying 指定LVS 的工作模式为直接路由模式（也是LVS 默认的模式）  
-i --ipip 指定LVS 的工作模式为隧道模式  
-m --masquerading 指定LVS 的工作模式为NAT 模式  
-w --weight weight 真实服务器的权值  
--mcast-interface interface 指定组播的同步接口  
-c --connection 显示LVS 目前的连接 如：ipvsadm -L -c  
--timeout 显示tcp tcpfin udp 的timeout 值 如：ipvsadm -L --timeout  
--daemon 显示同步守护进程状态  
--stats 显示统计信息  
--rate 显示速率信息  
--sort 对虚拟服务器和真实服务器排序输出  
--numeric -n 输出IP 地址和端口的数字形式

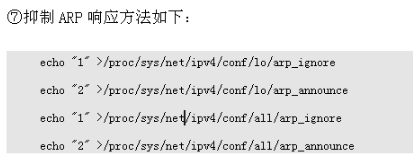
配置完了，检查方法：



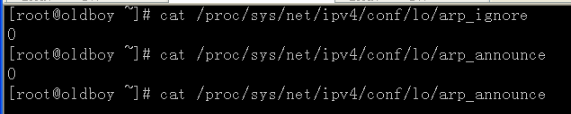
### RS服务器上绑定VIP

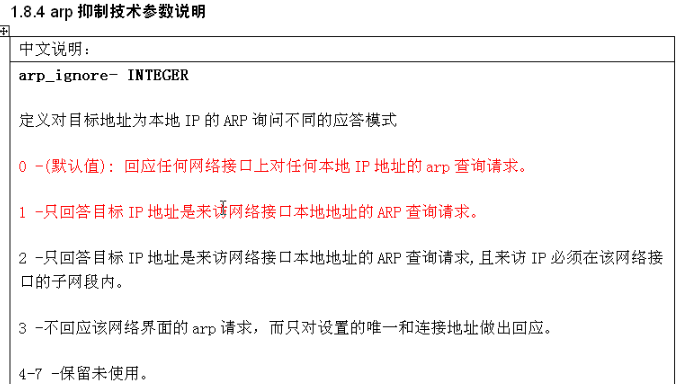
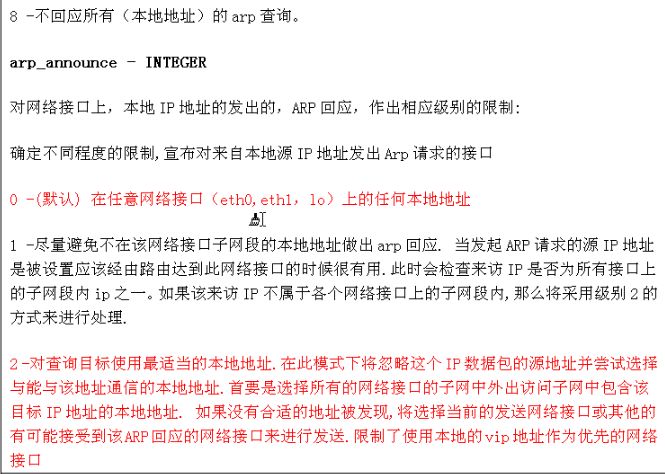


### 手工在RS服务器上抑制ARP



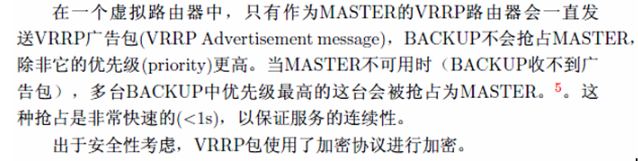
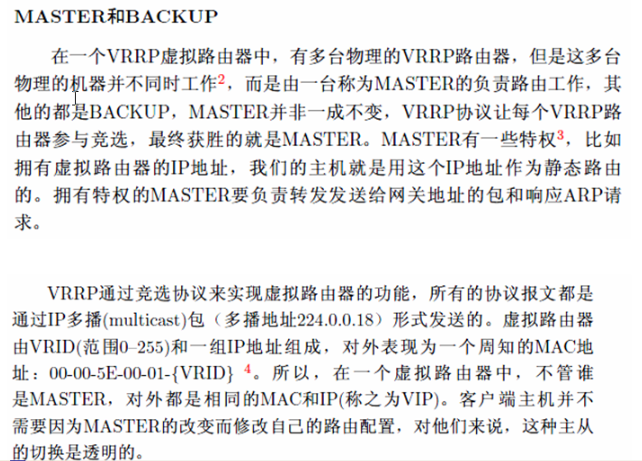
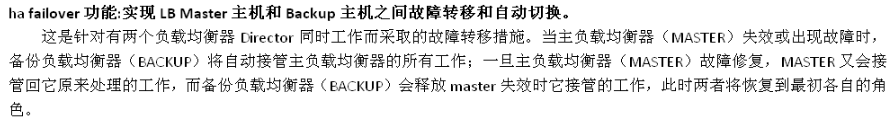
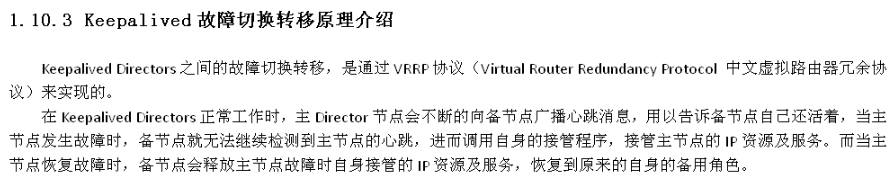
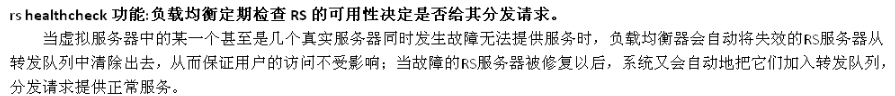
改之前可以用命令先查看之前内容





# Keepalived

Keepalived起初是为了LVS设计的，专门用来监控集群系统中各个服务节点的状态，后来加入了VRRR的功能，VRRR是虚拟路由器冗余协议，VRRR的出现的目的是为了解决静态路由出现的单点故障问题，它能够保证网络不间断、稳定的运行。所有keepalived一方面具有LVS cluster nodes healthchecks功能，一方面也有LVS directors failover功能。

## 安装配置

依赖包

yum install gcc openssl openssl-devel kernel-devel

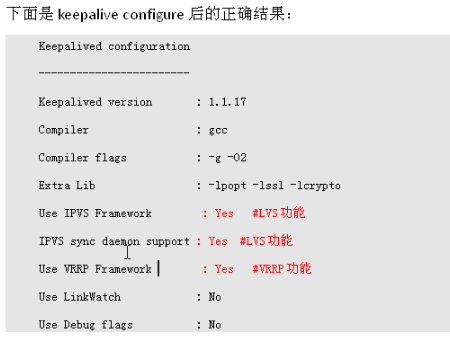
先做软连接，如果之前做了 可以不做



解压keepalived并进入目录



然后./configure



如果出现图上三个标红的是yes，表示安装成功。

最后make &&make install



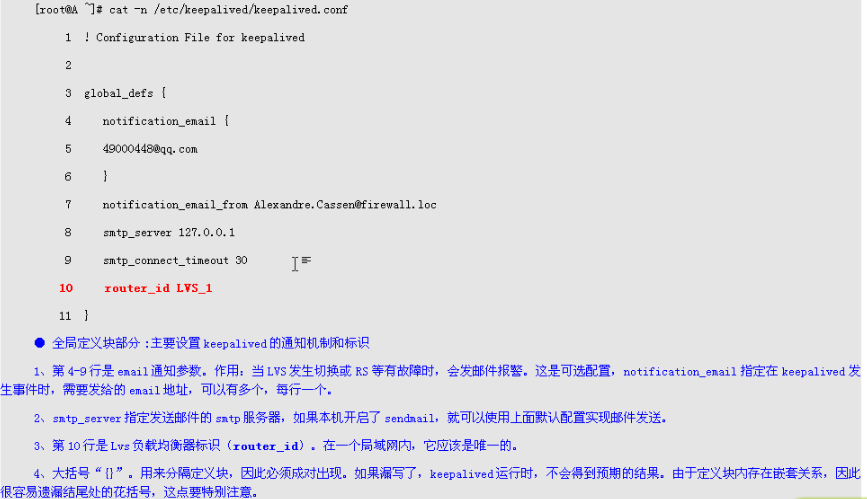
cp /usr/local/etc/rc.d/init.d/keepalived /etc/init.d/

cp /usr/local/etc/sysconfig/keepalived /etc/sysconfig/

cp /usr/local/etc/keepalived/keepalived.conf /etc/keepalived/

cp /usr/local/sbin/keepalived /usr/sbin

Keepalived配置文件中！表示注释



VRRR块定义

vrrp\_instance VI\_1 {

state MASTER

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

authentication { \\权限

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress { \\虚拟IP地址

192.168.200.16

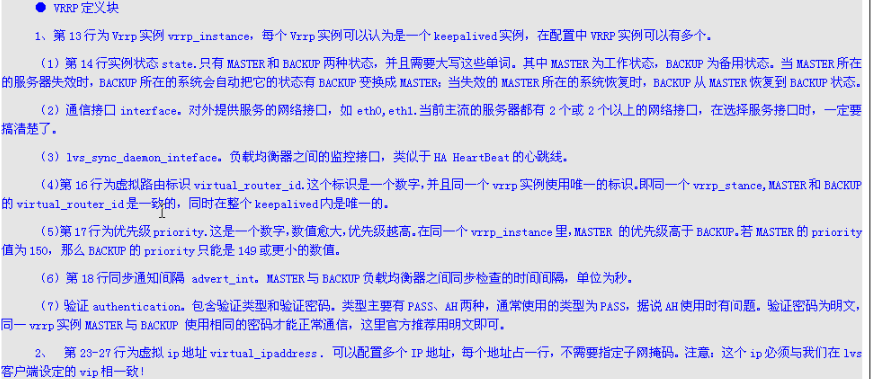
192.168.200.17

192.168.200.18

}

}

vrrp\_instance VI\_1表示一个实例



virtual\_router\_id为虚拟路由标识，这个标识是一个数字，并且同一个vrrp实例使用唯一一个标识，即同一个vrrp\_stance，master和backup的virtual\_router\_id是一致的，同时在整个keepalived内是唯一的。

advert\_int为同步通知间隔，master和backup负载均衡器之间同步检查的时间间隔，单位为秒。

virtual\_ipaddress为虚拟IP地址，可以配置多个IP地址，每个地址占一行，不需要指定子网掩码，但这个IP必须与lvs客户端设置的VIP一样。

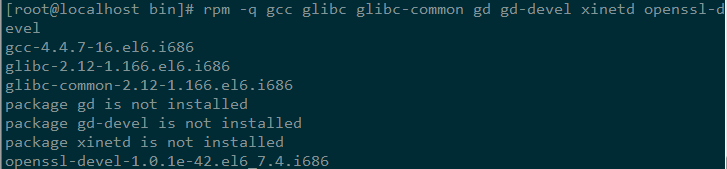
# Nagios

http://www.cnblogs.com/mchina/archive/2013/02/20/2883404.html

## 安装

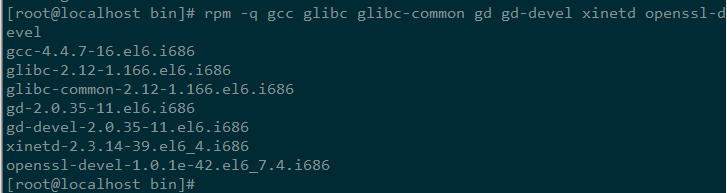
安装前检查依赖包

rpm -q gcc glibc glibc-common gd gd-devel xinetd openssl-devel



若有缺失,请先安装.





新增nagios账户，并指定家目录



创建一个用户组名为nagcmd用于从Web接口执行外部命令。将nagios用户和apache用户都加到这个组中。

[root@localhost ~]# groupadd nagcmd

[root@localhost ~]# usermod -G nagcmd nagios

解压nagios



1. cd nagios
2. ./configure --prefix=/usr/local/nagios
3. make all
4. make install && make install-init && make install-commandmode && make install-config

将nagios添加为服务

[root@localhost nagios]# service nagios status

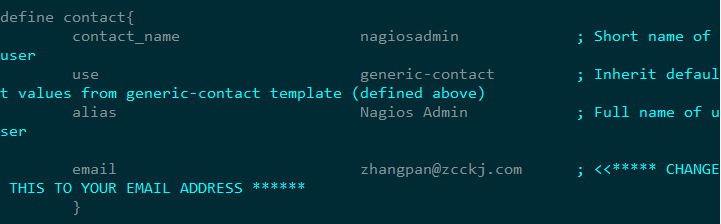
No lock file found in /usr/local/nagios/var/nagios.lock

[root@localhost nagios]# chkconfig --add nagios

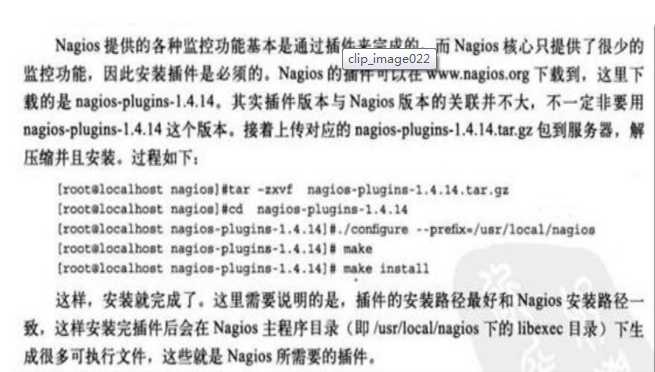
[root@localhost nagios]# chkconfig nagios on

[root@localhost nagios]# chkconfig --list nagios

nagios 0:off 1:off 2:on 3:on 4:on 5:on 6:off

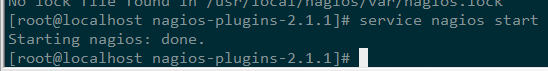
来编辑/usr/local/nagios/etc/objects/contacts.cfg配置文件，更改email地址nagiosadmin的联系人定义信息中的EMail信息为你的EMail信息以接收报警内容。   


安装nagios插件



启动nagios

Service nagios start



为了安全起见，一般情况下要让nagios 的web 监控页面必须经过授权才能访问，这需要增加验证配置，即在httpd.conf 文件最后添加如下信息：

<Directory "/usr/local/nagios/sbin">

AuthType Basic

Options ExecCGI

AllowOverride None

Order allow,deny

Allow from all

AuthName "Nagios Access"

AuthUserFile /usr/local/nagios/etc/htpasswd //用于此目录访问身份验证的文件

Require valid-user

</Directory>

Alias /nagios "/usr/local/nagios/share"

<Directory "/usr/local/nagios/share">

AuthType Basic

Options None

AllowOverride None

Order allow,deny

Allow from all

AuthName "nagios Access"

AuthUserFile /usr/local/nagios/etc/htpasswd

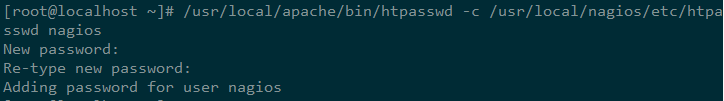
Require valid-user

</Directory>

创建apache目录验证文件

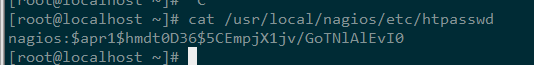
在上面的配置中，指定了目录验证文件htpasswd，下面要创建这个文件：

[root@localhost ~]# htpasswd -c /usr/local/nagios/etc/htpasswd nagios



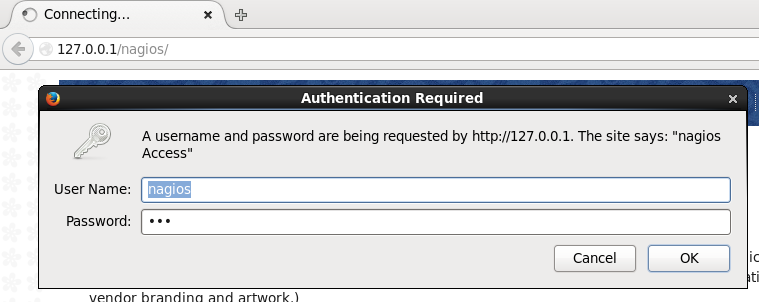
查看认证文件的内容

# cat /usr/local/nagios/etc/htpasswd



<http://127.0.0.1/nagios/>

访问时，输入用户名和密码：则是vim /usr/local/nagios/etc/htpasswd 该文件中的用户和密码！

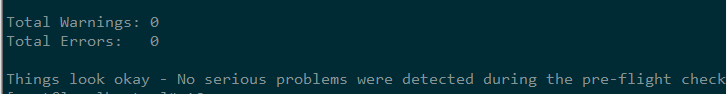


## 验证配置文件

[root@localhost ~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



如果结果最后显示都为0 表示安装正常



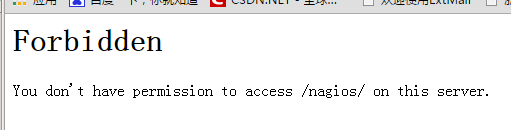
验证nagios配置语法是否错误

[root@localhost ~]# /etc/init.d/nagios checkconfig



## Nagios网页打开异常

如果打开网页出现如下问题,可能是没装PHP 也可



出现  The requested URL /nagios/cgi-bin/statusmap.cgi was not found on this server错误

解决方法：

确定安装gd-devel包

然后回到解压目录，

#make   clean

重新编译安装一下

# ./configure --with-gd-lib=/usr/lib --with-gd-inc=/usr/include

接下来

#./configure --with-command-group=nagcmd

#make all;

make install;

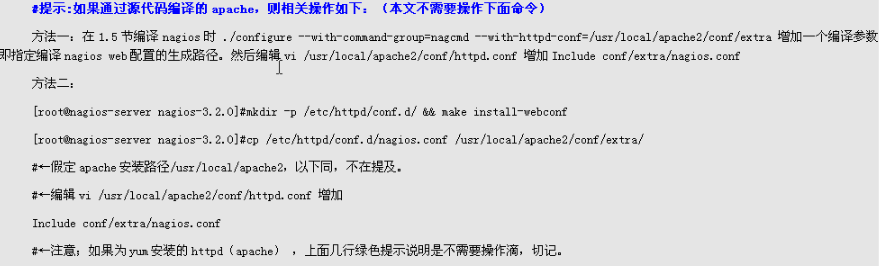
make install-init;

make install-config;

make install-commandmode;

make install-webconf

以上是因为没有安装apache,如果是编译安装的apache，则解决方法如下：

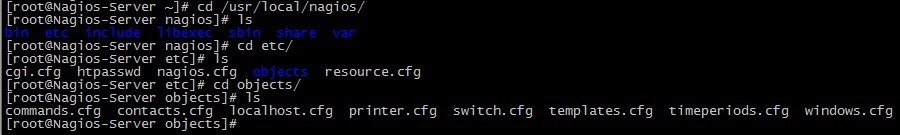


## 配置

Nagios 主要用于监控一台或者多台本地主机及远程的各种信息，包括本机资源及对外的服务等。默认的Nagios 配置没有任何监控内容，仅是一些模板文件。若要让Nagios 提供服务，就必须修改配置文件，增加要监控的主机和服务，下面将详细介绍。

默认配置文件介绍

Nagios 安装完毕后，默认的配置文件在/usr/local/nagios/etc目录下。

[](http://images.cnitblog.com/blog/370046/201301/30172103-011c4098757a465fa8e4462f89768c84.jpg)

每个文件或目录含义如下表所示：

|  |  |
| --- | --- |
| **文件名或目录名** | **用途** |
| cgi.cfg | 控制CGI访问的配置文件 |
| nagios.cfg | Nagios 主配置文件 |
| resource.cfg | 变量定义文件，又称为资源文件，在些文件中定义变量，以便由其他配置文件引用，如$USER1$ |
| objects | objects 是一个目录，在此目录下有很多配置文件模板，用于定义Nagios 对象 |
| objects/commands.cfg | 命令定义配置文件，其中定义的命令可以被其他配置文件引用 |
| objects/contacts.cfg | 定义联系人和联系人组的配置文件 |
| objects/localhost.cfg | 定义监控本地主机的配置文件 |
| objects/printer.cfg | 定义监控打印机的一个配置文件模板，默认没有启用此文件 |
| objects/switch.cfg | 定义监控路由器的一个配置文件模板，默认没有启用此文件 |
| objects/templates.cfg | 定义主机和服务的一个模板配置文件，可以在其他配置文件中引用 |
| objects/timeperiods.cfg | 定义Nagios 监控时间段的配置文件 |
| objects/windows.cfg | 监控Windows 主机的一个配置文件模板，默认没有启用此文件 |

6.2 配置文件之间的关系

在nagios的配置过程中涉及到的几个定义有：主机、主机组，服务、服务组，联系人、联系人组，监控时间，监控命令等，从这些定义可以看出，nagios各个配置文件之间是互为关联，彼此引用的。

成功配置出一台nagios监控系统，必须要弄清楚每个配置文件之间依赖与被依赖的关系，最重要的有四点：

第一：定义监控哪些主机、主机组、服务和服务组；   
第二：定义这个监控要用什么命令实现；   
第三：定义监控的时间段；   
第四：定义主机或服务出现问题时要通知的联系人和联系人组。

6.3 配置Nagios

 为了能更清楚的说明问题，同时也为了维护方便，建议将nagios各个定义对象创建独立的配置文件：

* 创建hosts.cfg文件来定义主机和主机组
* 创建services.cfg文件来定义服务
* 用默认的contacts.cfg文件来定义联系人和联系人组
* 用默认的commands.cfg文件来定义命令
* 用默认的timeperiods.cfg来定义监控时间段
* 用默认的templates.cfg文件作为资源引用文件

在主配置文件nagios.cfg中，可以新增主机和服务器配置文件，如：

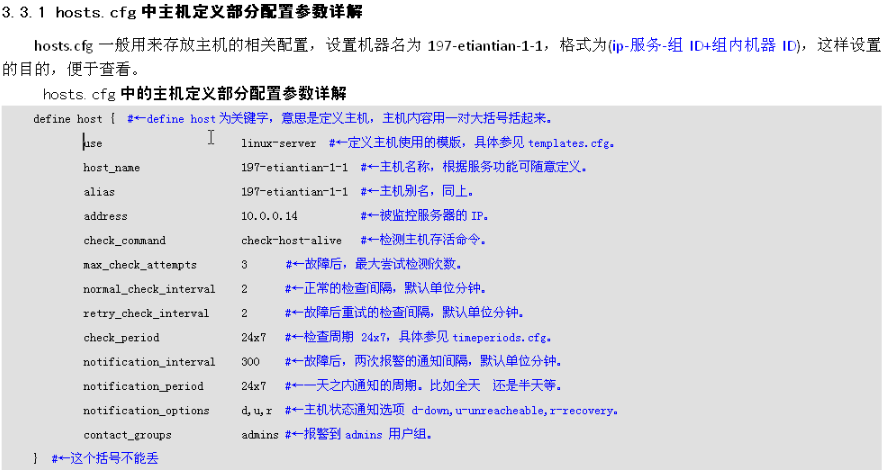
cfg\_file=/usr/local/nagios/etc/objects/hosts.cfg

cfg\_file=/usr/local/nagios/etc/objects/servers.cfg

也可用cfg\_dir来新增目录，在新增的目录下，只要文件符合\*.cfg都可以被nagios加载

注释掉 cfg\_file=/usr/local/nagios/etc/objects/localhost.cfg这一行

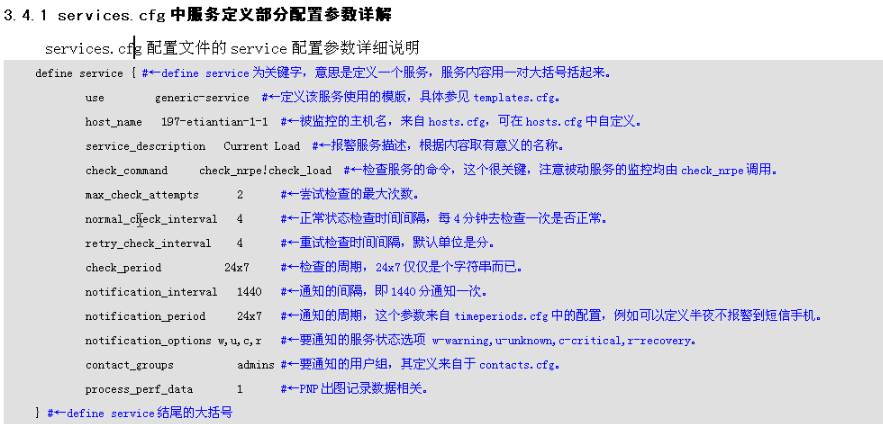
Localhost.cfg是nagios服务器监控本地服务的配置，注释掉之后，可以统一进行监控



head -51 localhost.cfg >hosts.cfg

把localhost.cfg文件中前51行复制到hosts.cfg中，如果hosts.cfg文件不存在，则新建hosts.cfg。

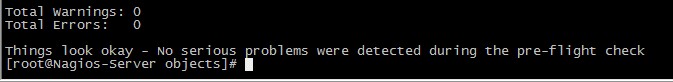
Services.cfg文件是配置监控服务的，是nagios最重要的配置文件之一，这个配置文件默认不存在，需要手工添加。



6.4 验证Nagios 配置文件的正确性

Nagios 在验证配置文件方面做的非常到位，只需通过一个命令即可完成：

# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



Nagios提供的这个验证功能非常有用，在错误信息中通常会打印出错误的配置文件以及文件中的哪一行，这使得nagios的配置变得非常容易，报警信息通常是可以忽略的，因为一般那些只是建议性的。

看到上面这些信息就说明没问题了，然后启动Nagios 服务。