GNU/Linux- 負載均衡
HAProxy

GNU/Linux-HAProxy

HAProxy 提供高可用性、负载均衡以及基于 TCP 和 HTTP 应用的代理，持虚拟主机它是免费、快速并且可靠的一种解决方案

HAProxy 特别适用于那些负载特大的 web 站点，这些站点通常又需要会话保持或七层处理。

GNU/Linux-HAProxy

HAProxy 运行在当前的硬件上，可以支持数以万计的并发连接并且它的运行模式使得它可以很简单安全的整合进您当前的架构中时可以保护你的 web 服务器不被暴露到网络上.

GNU/Linux-HAProxy

HAProxy 实现了一种事件驱动，单一进程模型，此模型支持非常大的并发连接数。

多进程或多线程模型受内存限制统调度器限制，以及无处不在的锁限制，很少能处理数千并发连接。

GNU/Linux-HAProxy

事件驱动模型因为在有更好的资源和时间管理的用户空间 (User-Space) 实现所有这些任务，所以没有这些问题。

此模型的弊端是，在多核系统上，这些程序通常扩展性较差。这就是为什么他们必须进行优化以使每个 CPU 时间片 (Cycle) 做更多的工作。

GNU/Linux-HAProxy

HAProxy
一:HTTP 负载均衡(L7)
1. 安装 HAProxy
#yum install haproxy -y

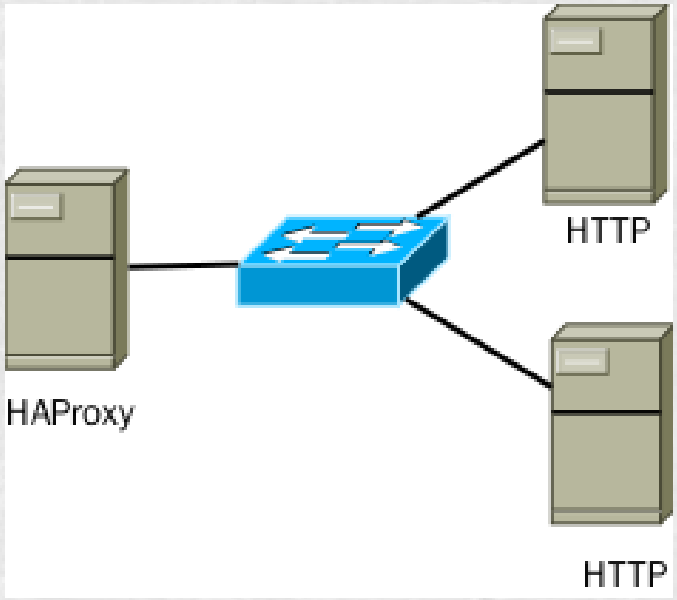
2. 配置 HAproxy
#cd /etc/haproxy
#mv haproxy.cfg haproxy.cfg.bak
#vi haproxy.cfg

GNU/Linux-HAProxy

HAProxy

一 :HTTP 负载均衡 (L7)

示意图



GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

Global

//* 设定 HAproxy 的日志于 info 以上信息将被记录在本地的 local2 上

log 127.0.0.1 local2 info

chroot /var/lib/haproxy

pidfile /var/run/haproxy.pid

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

//* 最大连接数为 256
maxconn 256

//* 指定 haproxy 进程的属主或数组
user haproxy
group haproxy

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy
   //*haproxy 在系统后台运行
   daemon

defaults
   //* 运行模式为 http
   mode http

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

//* 日志参考 global

log global

//* 获取 http 请求保存至日志

option httplog

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

```
//* 后台连接超时时间
timeout connect 10s
//* 客户端连接超时时间
timeoute client 30s
//* 服务器连接超时时间
timeout server 30s
```

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

//* 设定 HA 所需的内容 ,http-in 为 frontend 指定的名称
frontend http-in
    //* 设定 HAProxy 所监听的端口
    Bind *:80

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

```
//* 设定默认后台名称为 niliu_server
default_backend niliu_server

//* 允许后端服务器记录客户端的 IP 地址
option forwardfor
```

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

//* 关于 option

option originalto ：如果服务器上的应用程序想记录发起请求的原目的 IP 地址，需要在 HAProxy 上配置此选项，这样 HAProxy 会添加 "X-Original-To" 字段。

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

//* 关于 option

option dontlognull ：保证 HAProxy 不记录上级负载均衡发送过来的用于检测状态没有数据的心跳包。

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

//* 关于 option

option forwardfor ：如果服务器上的应用程序想记录发起请求的客户端的 IP 地址，需要在 HAProxy 上配置此选项，这样 HAProxy 会把客户端的 IP 信息发送给服务器 ,HTTP 请求中添加 "X-Forwarded-For" 字段。

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

//* 关于 option

option httpclose :HAProxy 会针对客户端的第一条请求的返回添加 cookie 并返回给客户端，客户端发送后续请求时会发送此 cookie 到 HAProxy ， HAProxy 会针对此 cookie 分发到上次处理此请求的服务器上，果服务器不能忽略此 cookie 值会影响处理结果，免这种情况配置此选项，防止产生多余的 cookie 信息。

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy

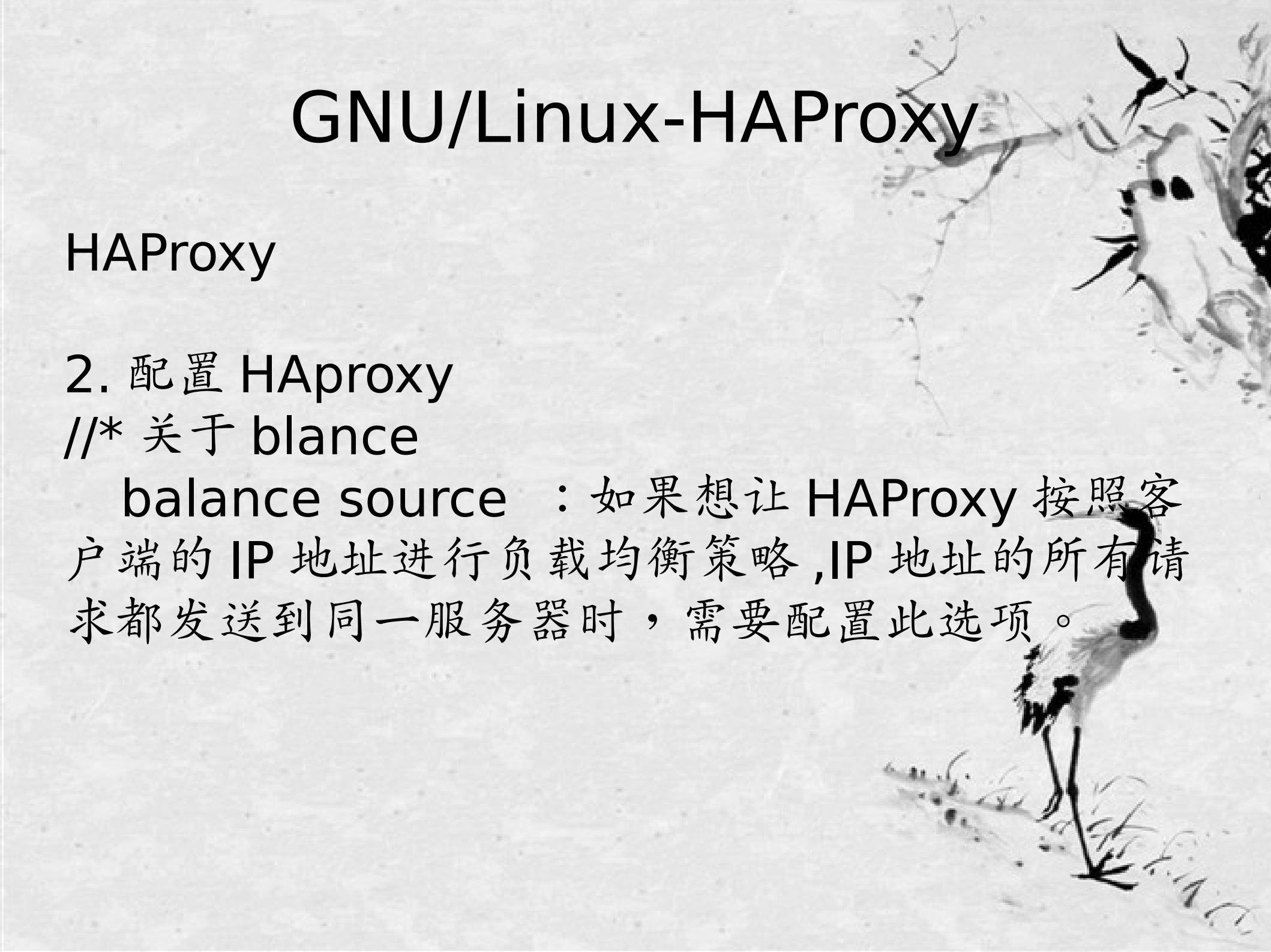
//* 设定后台集群

```
backend niliu_servers
    balance roundrobin
    server n1.niliu.edu 192.168.188.12:80
check
    server n2.niliu.edu 192.168.188.13:80
check
#systemctl start haproxy
#systemctl enable haproxy
```

GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy
//* 关于 blance
　balance source ：如果想让 HAProxy 按照客户端的 IP 地址进行负载均衡策略 ,IP 地址的所有请求都发送到同一服务器时，需要配置此选项。



GNU/Linux-HAProxy

HAProxy

2. 配置 HAproxy
//* 关于 balance
　balance roundrobin:HAProxy 把请求轮流的转发到每一个服务器上，台服务器的权重，此权重会动态调整，最常见的默认配置

GNU/Linux-HAProxy

HAProxy

3. 配置 rsyslog.conf

#vi /etc/rsyslog.conf

解开 15,16 行注释，并添加 17 行
$ModLoad imudp
$UDPServerRun 514
$AllowedSender UDP, 127.0.0.1

GNU/Linux-HAProxy

HAProxy

3. 配置 rsyslog.conf
在修改 55 行增加
*.info;mail.none;authpriv.none;cron.none;local2.none　/var/log/messages
//*56 行增加
 local2.* /var/log/haproxy.log
#systemctl restart rsyslog

GNU/Linux-HAProxy

HAProxy

4. 配置 n1/n2 两台 web 服务器的 httpd.conf

#vi /etc/httpd/conf/httpd.conf
//*196 行修改为
LogFormat "\"%{X-Forwarded-For}i\" %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
#systemctl restart httpd

GNU/Linux-HAProxy

HAProxy

5. 浏览器测试

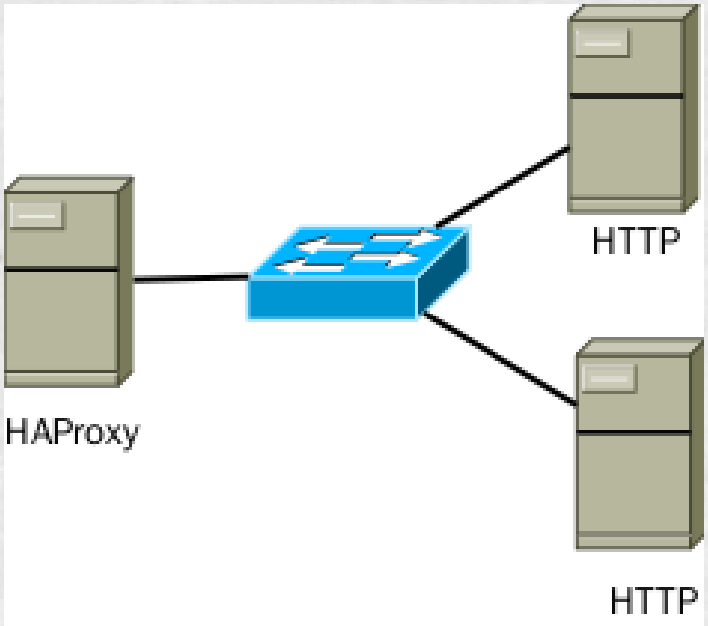
通过访问 HAProxy 可以访问到后端 HTTP 服务器

GNU/Linux-HAProxy

HAProxy

二：在 HA 上设定 SSL

示意图：



GNU/Linux-HAProxy

HAProxy

二：在 HA 上设定 SSL

1) 在 HAProxy 上创建私钥和证书

```
#cd /etc/pki/tls/certs
# openssl req -x509 -nodes -newkey rsa:2048 -keyout /etc/pki/tls/certs/haproxy.pem -out /etc/pki/tls/certs/haproxy.pem -days 365
```

GNU/Linux-HAProxy

HAProxy

二：在 HA 上设定 SSL

1) 在 HAProxy 上创建私钥和证书
#chmod 600 haproxy.pem

GNU/Linux-HAProxy

HAProxy

二：在 HA 上设定 SSL
2) 修改 HAProxy 配置文件
#vi /etc/haproxy/haproxy.cfg
//* 在 global 下增加如下内容
//* 指定 ssl 虽大连接数
Maxsslconn 256
//* 指定秘钥类型及长度
tune.ssl.default-dh-param 2048

GNU/Linux-HAProxy

HAProxy

二：在 HA 上设定 SSL
2) 修改 HAProxy 配置文件
#vi /etc/haproxy/haproxy.cfg
//* 在 frontend http-in 区段增加指定的监听端口及认证证书 (以下是一行)
bind *:443 ssl crt /etc/pki/tls/certs/haproxy.pem

GNU/Linux-HAProxy

HAProxy

二：在 HA 上设定 SSL
3) 重启 haproxy
#systemctl restart haproxy

4) 使用浏览器方位
https://haproxy_server_ip

GNU/Linux-HAProxy

HAProxy

三：监控 HAProxy 状态
1) 修改 HAProxy 配置文件
#vi /etc/haproxy/haproxy.cfg

//* 在 frontend http-in 区段下增加如下内容
//* 开启状态报告
stats enable

GNU/Linux-HAProxy

HAProxy

三：监控 HAProxy 状态
1) 修改 HAProxy 配置文件
//* 设定进入 web 报表的账户名及密码
stats auth snow:5iblue.com.cn

//* 隐藏 haproxy 版本信息
stats hide-version

GNU/Linux-HAProxy

HAProxy

三：监控 HAProxy 状态
1) 修改 HAProxy 配置文件

//* 显示 HAProxy 主机名
stats show-node

//* 设定报告刷新时间
stats refresh 60s

GNU/Linux-HAProxy

HAProxy

三：监控 HAProxy 状态
1) 修改 HAProxy 配置文件

//* 设定报告的路径
stats uri /haproxy?stats

#systemctl restart haproxy

GNU/Linux-HAProxy

HAProxy

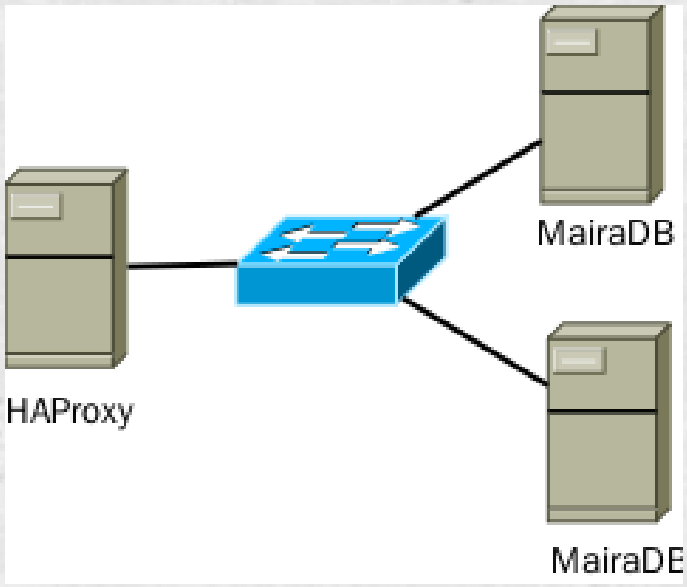
三：监控 HAProxy 状态
2) 测试
http://haproxy_srv_ip/haproxy?stats

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

示意图



GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡
0) 在所有后端安装 MariaDB
#yum install mariadb-server

#vi /etc/my.cnf
//* 在 [mysqld] 区段最后设定所支持的字符集
character-set-server=utf8

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡
1) 在所有后端安装 MariaDB
systemctl start mariadb
systemctl enable mariadb

2) 初始化 Mariadb
mysql_secure_installation

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡
设定数据库 root 的密码
Set root password? [Y/n] y

移除匿名账户
Remove anonymous users? [Y/n] y

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

禁止 root 账户远程登录
Disallow root login remotely? [Y/n] y

移除测试数据库
Remove test database and access to it? [Y/n] y

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

重新加载权限表
Reload privilege tables now? [Y/n] y

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

3) 登录测试

```
# mysql -u root -p
> select user,host,password from
mysql.user;
> show databases;
> exit
```

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

4) 为所有后端创建数据库
Node1
#mysql -u root -p
>create database test1;

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

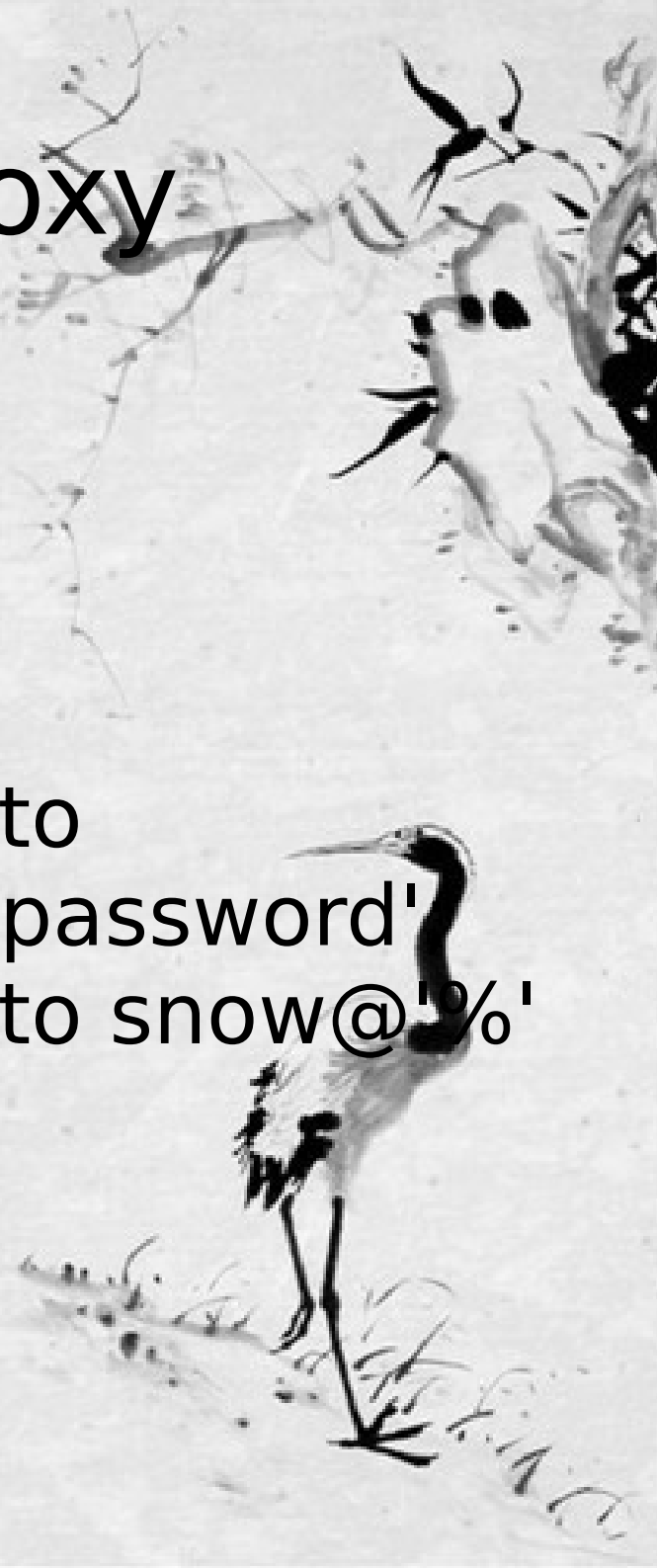
4) 为所有后端创建数据库并授权

>grant all privileges on test1.* to snow@'localhost' identified by 'password'

>grant all privileges on test1.* to snow@'%' identified by 'password';

> flush privileges;

>exit



GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

4) 为所有后端创建数据库
Node2
#mysql -u root -p
>create database test2;

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

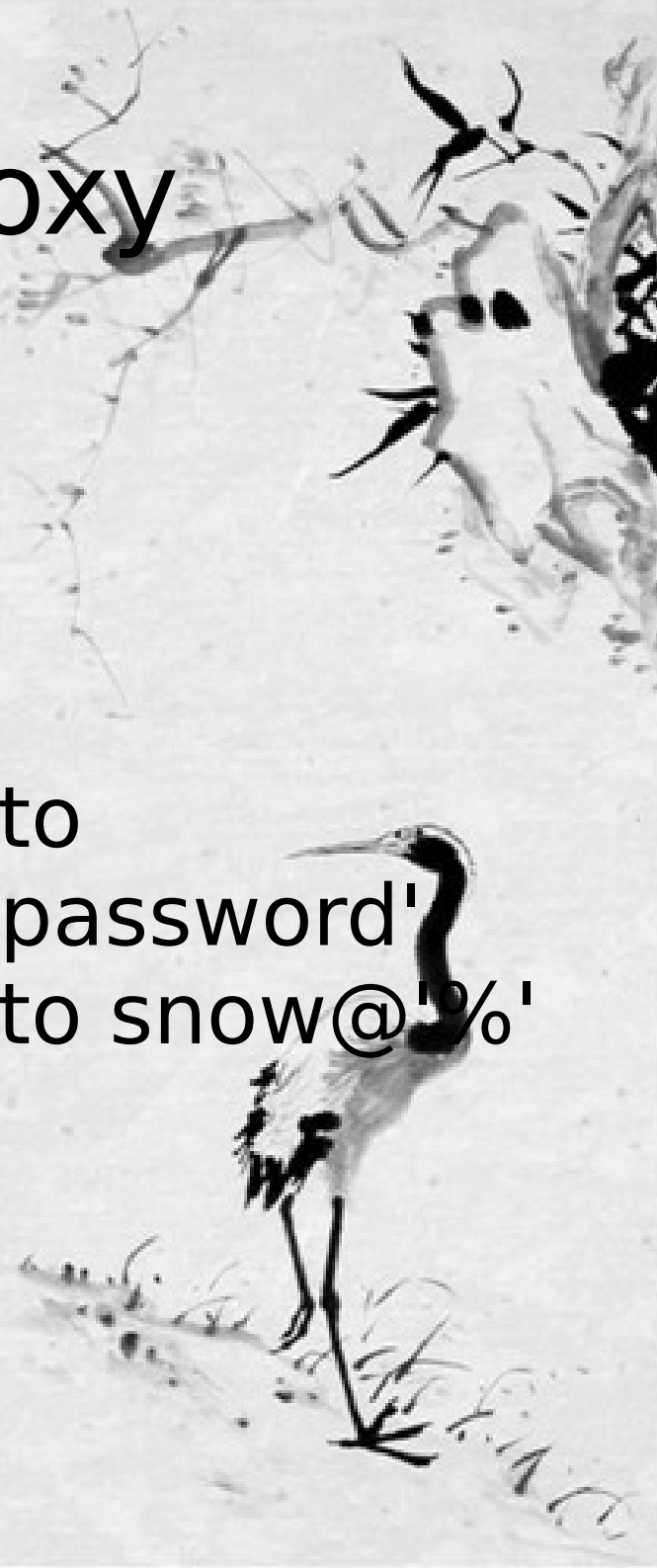
4) 为所有后端创建数据库并授权

>grant all privileges on test2.* to snow@'localhost' identified by 'password'

>grant all privileges on test2.* to snow@'%' identified by 'password';

> flush privileges;

>exit



GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

4) 为所有后端创建数据库
Node3
#mysql -u root -p
>create database test3;

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

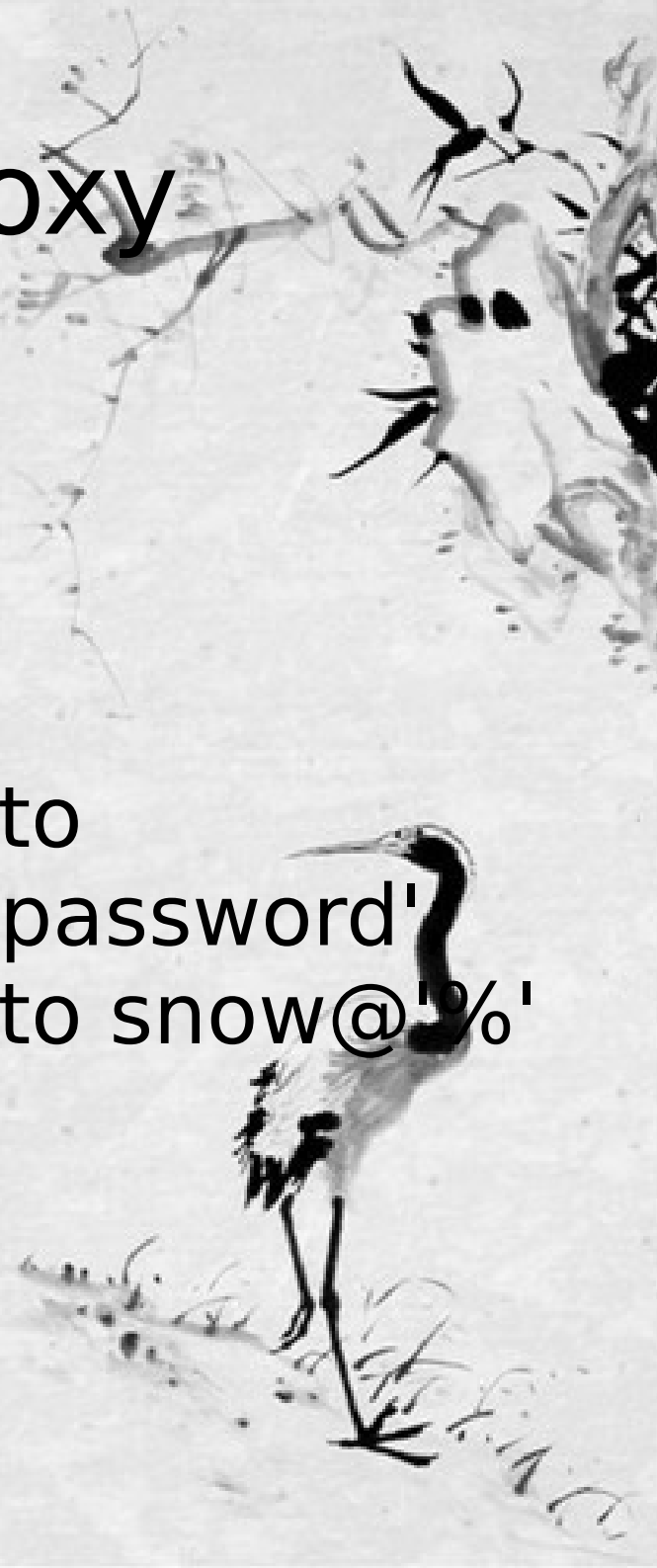
4) 为所有后端创建数据库并授权

>grant all privileges on test3.* to snow@'localhost' identified by 'password'

>grant all privileges on test3.* to snow@'%' identified by 'password';

> flush privileges;

>exit



GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

5) 编辑 HAProxy

```
# vi /etc/haproxy/haproxy.cfg
global
    log         127.0.0.1 local2 info
   chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
```

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

5) 编辑 HAProxy

```
maxconn     256
maxsslconn  256
user        haproxy
group       haproxy
daemon
```

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

5) 编辑 HAProxy

```
defaults
    //* 设定 L4 关键点 , 模式采用 TCP
    mode                    tcp
    log                     global
    timeout connect    10s
    timeout client     30s
    timeout server     30s
```

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

5) 编辑 HAProxy

frontend mysql-in

    //*mariadb 所监听端口为 3306. 因此 haproxy 也监听此端口

    bind *:3306

    default_backend   backend_mariadb

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

5) 编辑 HAProxy

```
backend backend_servers
    balance          roundrobin
    server           n1 192.168.188.12:3306
check
    server           n2 192.168.188.13:3306
check
```

```
# systemctl restart haproxy
```

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

6) 测试

#mysql -u snow -p -h 192.168.188.11

Enter password:password

>show databases;

>exit

GNU/Linux-HAProxy

HAProxy

四 :HAProxy 之 L4 负载均衡

//*L4 负载仅仅是以 MariaDB 作为例子 . 在实际生产过程中应该所有的后端 MariaDB 服务器的数据库及表、键等都是一致的。可通过 MariaDB 的数据库同步达到一致性