**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*iptables\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

RHEL7里面同时有firewall和iptables

从firewall里面设置的防火墙规则，会被自动转换为iptables规则

firewall

iptables

内核（netfilter）

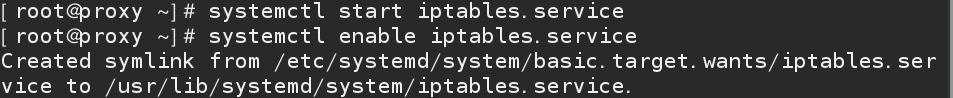
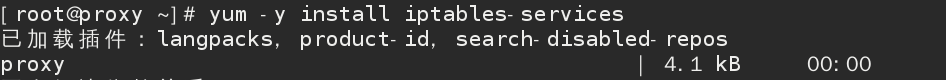
生产环境中很多企业还是在使用iptables

比如：Openstark,Openshift,Docker,k8s项目

Openstack(kvm,vmware虚拟化)(一个机房500台主机，每台开10个虚拟机：快速启动、主机故障、虚拟机迁移、性能动态变化（根据性能动态调整虚拟机）)(Openstack Web点击鼠标，启动虚拟机)

Openshift，k8s(Docker集群管理)

iptables工作在网络层，是网络层防火墙



**关闭firewall**

**开启iptables**

**\*\*\*\*\*\*\*iptables有四表五链**

**四表：filter,nat,raw,mangle**

**【filter过滤表：设置允许和拒绝的规则】【重点】**

**【nat地址转换表：内网IP转外网IP，外网IP转内网IP】【重点】**

【raw数据表跟踪表：记录每个人的连接】【工作中推荐关闭raw数据表跟踪表，保持性能】

【mangle修改状态信息：可以修改包头，做伪装】【工作中几乎不用】

**五链：INPUT，OUTPUT，FORWARD，PREROUTING，POSTROUTING**

【INPUT：入站规则】

【OUTPUT：出站规则】

【FORWARD：转发规则：软路由时应用】

【PREROUTING：路由前规则：软路由时应用】

【POSTROUTING:路由后规则：软路由时应用】

**规则要写在某个表中的某个链**

**\*\*\*\*命令的语法格式**

**iptables [-t 表名] 选项[链名] [条件] [-j 目标操作]**

**选项：**

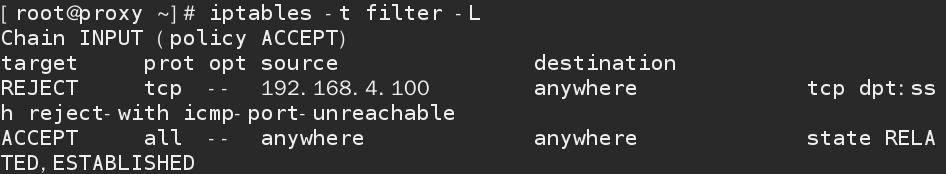
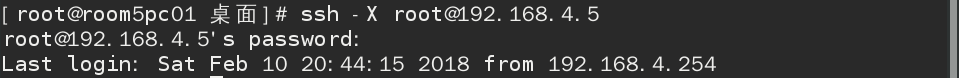
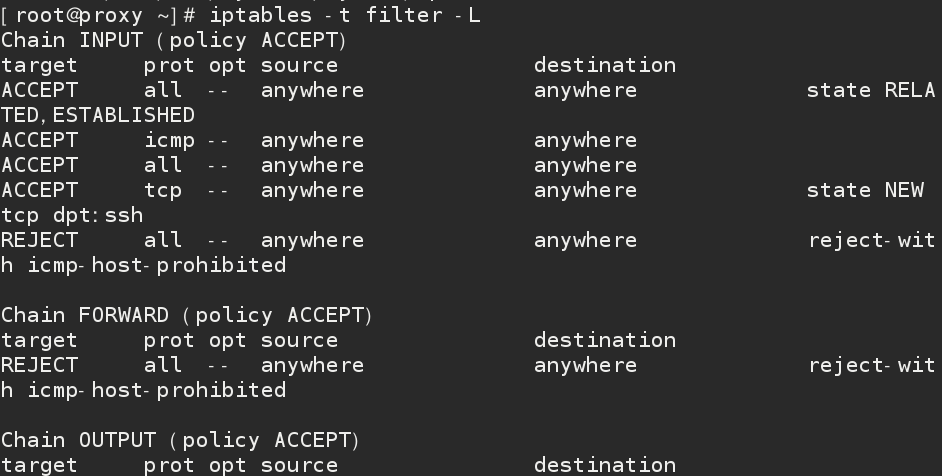
**-A：追加规则**

**-I：插入规则**

**-L：查看规则**

**-D：删除规则**

**j的目标操作:REJECT, DROP, ACCEPT**



**如果源地址是192.168.4.100，使用tcp协议，访问我的80端口，则接受**

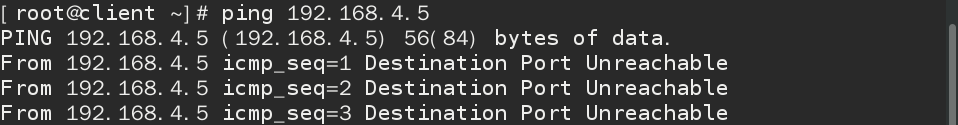
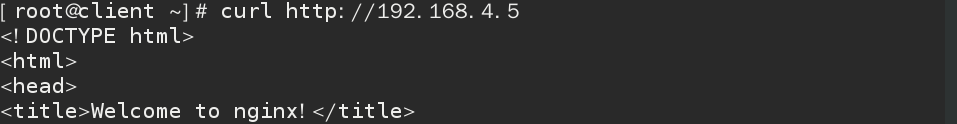
**查看，此时添加的规则在最上面**

**真实机则可以远程4.5**

**4.100不可以远程4.5**

**如果源地址是192.168.4.100，使用tcp协议，访问我的22端口，则拒绝**

**#查看规则**



**如果源地址是192.168.4.100，使用icmp协议，则接受**

**此时可以访问4.5的80端口**

**注意事项/整体规律：**

-可以不指定表，默认是filter表

-可以不指定链，默认对应表的所有链（一般都要指定）

-除非设置默认策略，否则必须指定匹配条件（有规则先匹配规则，没有规则匹配默认规则）

-选项/链名/目标操作用大写字母，其余都小写



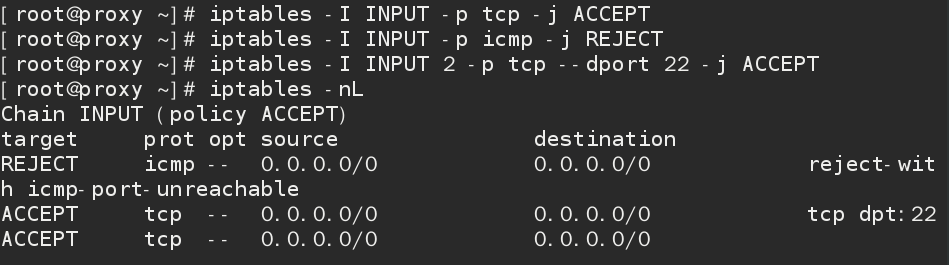
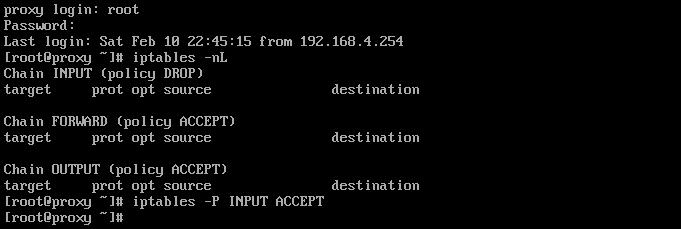
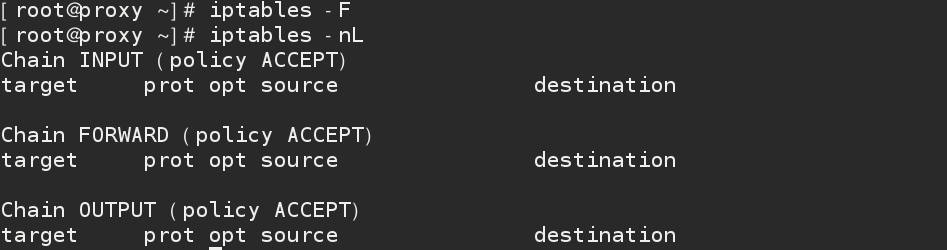
如果有人远程本机的ssh服务，则记录连接信息到/var/log/messages

**LOG**:记录日志，然后传给下一条规则，不满足“匹配即停止”

**查看filter所有链规则**



**查看filter所有链规则number**



**iptables -t nat -F //清空nat表**

**iptables -t raw -F //清空raw表**

**iptables -t mangle -F //清空mangle表**

**仅把filter里面的所有链清空**

**清空所有规则（仅对filter操作），之后默认允许所有**

**追加到最后一行**

**插入到第二行**

**设置默认规则**

**删除第三条规则**

**查看filter所有链规则number,加行号**

**\*\*\*\*\*\*linux软路由**

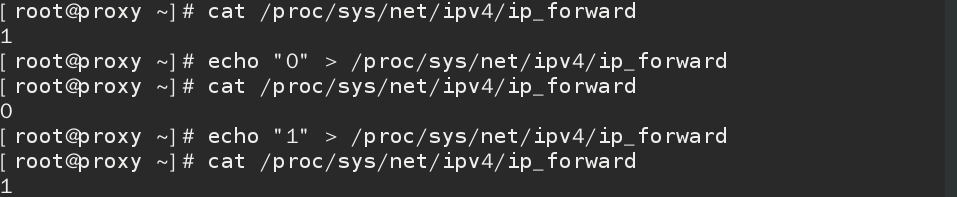
**在192.168.4.5主机上开启路由转发功能（路由器）**

web1:2.100

proxy:4.5/2.5

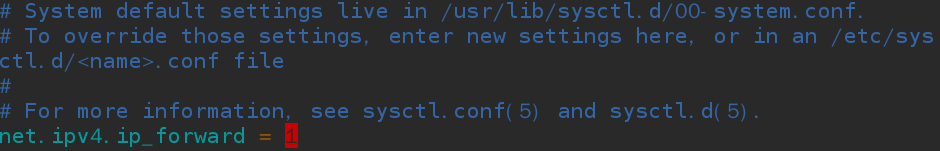
client:4.100

web2:2.200



**开启路由转发**

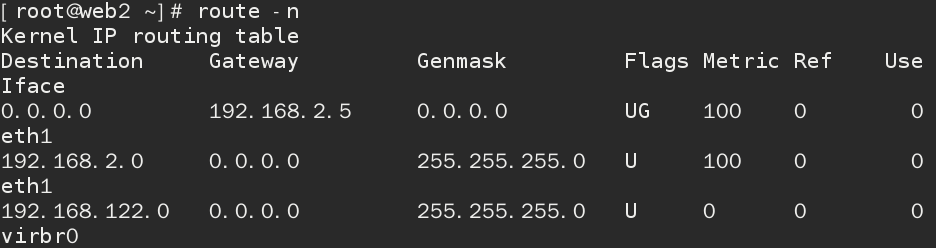
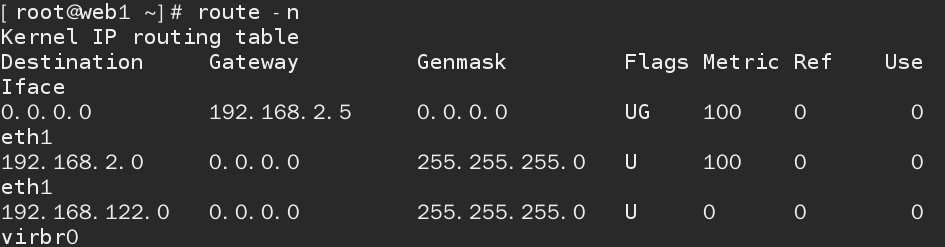
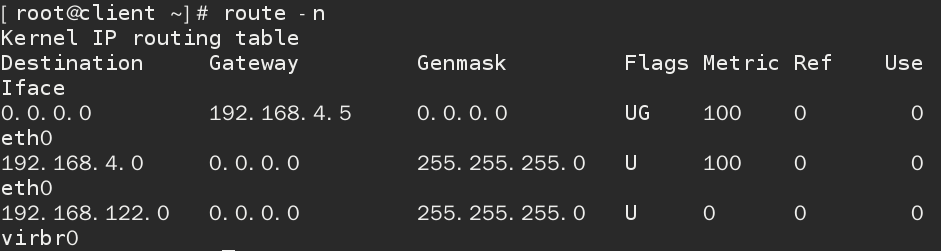
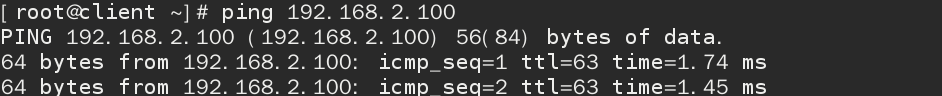
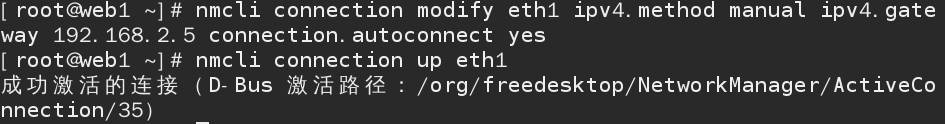
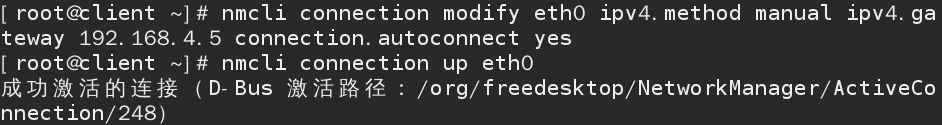
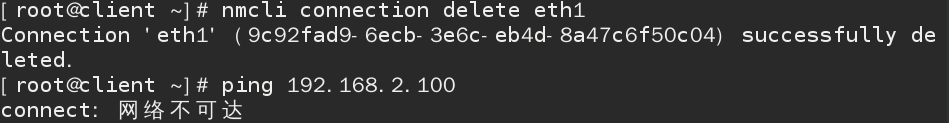
**关闭路由转发**



**查看内核参数**

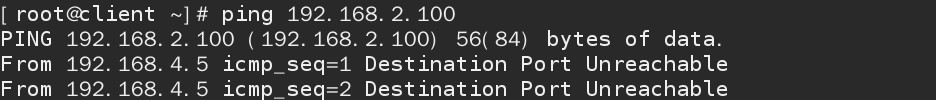
**永久开启路由转发**

给4.0网段的主机和2.0网段的主机添加网关，即可实现不同网络互联



**查看路由表**

**如果有人让路由转发icmp的包，则拒绝**



**如果通过路由远程内网里的主机ssh，则拒绝**



**\*\*\*\*\*\*iptables的条件规则**

**-s 匹配源ip地址**

**-d(--dst) 匹配目标ip地址**

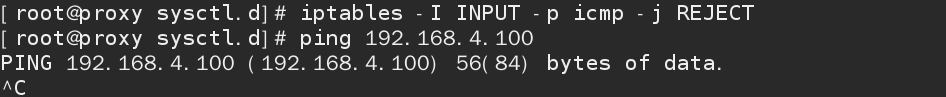
**--sport 匹配源端口**

**--dport 匹配目标端口**

**-i 匹配进站网卡**

**-o 匹配出站网卡**

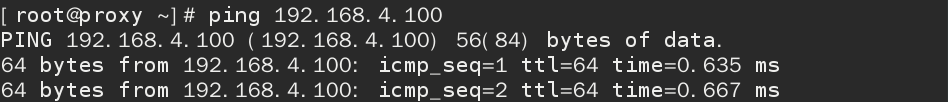
**--icmp-type 匹配icmp类型**



**其他主机无法ping本机，同时本机也无法ping其他主机**

**除4.100外，使用tcp访问本机的1-1024端口都拒绝**

！取反，！ -s 192.168.4.100 是！的有效范围，这条规则不决定4.100是否访问本机



**可以查看icmp的帮助**

**可以ping 别人，但是别人不可以ping 本机**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*iptables扩展过滤\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**iptables [-t 表名] 选项[链名][条件][-j 目标操作]**

iptables扩展过滤格式：

-m 扩展模块 --扩展条件 条件值

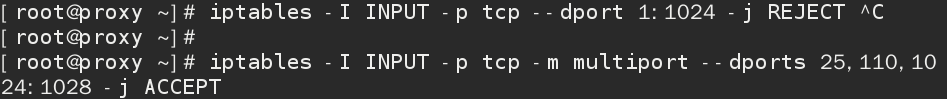
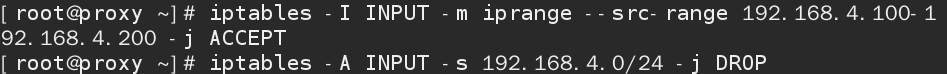
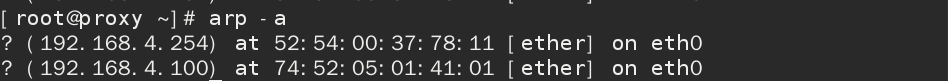
-m mac --mac-source MAC地址

-m multiport --sports 源端口列表

-m multiport --dports 目标端口列表

-m iprange --src-range IP1-IP2

-m iprange --dst-range IP1-IP2



**一次性允许多个端口，连续的用冒号隔开，不连续的用逗号隔开**

**一次性允许多个端口，必须是连续的**

**允许4.100-4.200，其他4.0网段ip都丢弃**

**可以查看与自己通信的ip 和mac地址**

**限制IP，但对方修改IP后，可以跳过规则，限制MAC地址，修改IP也不会跳过该规则**

参考帮助：man iptables

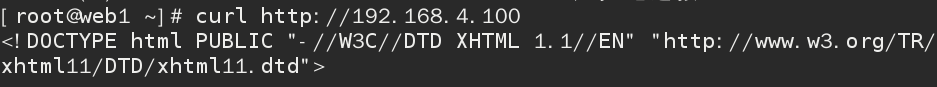
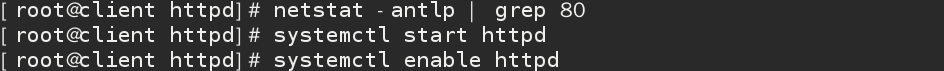
man extensions

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*NAT地址转换\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SNAT：源地址转换（内网主机需要访问外网）**

**内网的主机是私有IP，外网是公有IP**

**默认如果路由器仅开启转发功能，路由会转发数据包，但是，不会修改包头里面的源IP和目标IP地址，当使用私有IP访问公网的web时，查看日志，看谁在访问自己，会发现时一个私有IP。下一步，这个网站回应数据，回应给私有IP(私有IP不能在公网使用)**



**#web1:2.100**

DNAT目标地址转换（让公网可以访问私网）

