**一、使用w查看系统负载**

14:42:49 up 15 min(登录系统的时间), 1 user, load average: 0.00, 0.04, 0.08

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

root pts/0（网络登录） 192.168.1.19 14:28 1.00s 0.06s 0.02s w

2、load average: 0.00, 0.04, 0.08 //系统1分钟占用CPU的进程数的值、5分钟、15分钟的负载

3、cat /proc/cpuinfo //0代表系统有1个逻辑CPU，负载值不超过1就很好

cat /proc/cpuinfo

processor : 0

4、uptime //这个命令类似于w的第一行

15:16:43 up 48 min, 1 user, load average: 0.00, 0.01, 0.05

5、vmstat

r b swpd free buff cache si so bi bo in cs us sy id wa st

1 0 0 168808 876 338460 0 0 45 3 55 72 1 0 98 0 0

0 0 0 168828 876 338460 0 0 0 0 83 133 0 0 100 0 0

0 0 0 168828 876 338460 0 0 0 0 58 105 0 0 100 0 0

0 0 0 168828 876 338460 0 0 0 12 71 114 0 1 100 0 0

0 0 0 168828 876 338460 0 0 0 0 72 125 0 0 100 0 0

r:正在排队等待使用CUP的进程数、和正常使用CPU的进程数

b:等待资源的进程数

si:有多少数据从swp进入到内存中

so:从内存进入swp的数据

bi：=block in，表示从块设备读取数据（写入内存）的量（即：读磁盘）

bo：=block out，表示（从内存读取）写入到块设备的数据（即：写磁盘）

us:用户态的资源占用CUP的百分比

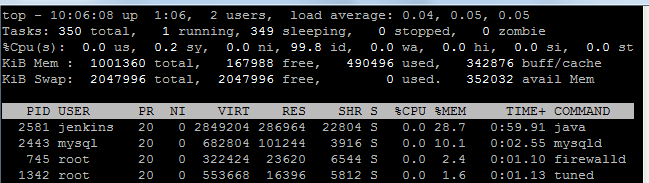
sy:系统本身的一些进程占用CPU的百分比

id:空闲的cpu百分比

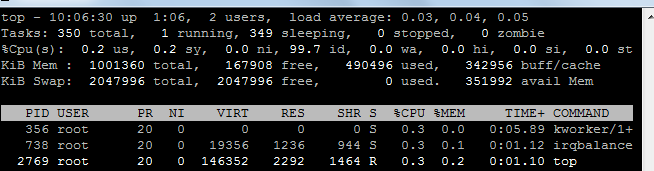
wa:等待cpu进程的百分比

**top命令**

M:按照内存的使用从大到小排序

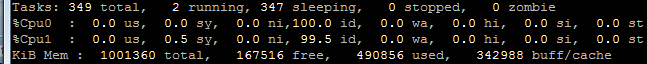


P：按照CPU的使用从大到小排序



RES :使用的物理内存数量，单位为Kib

"1":显示所有CPU使用情况



top -bn1 //静态显示状态

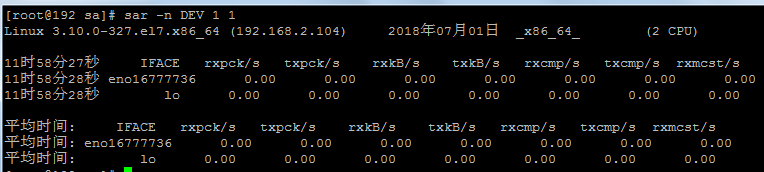
**sar命令**

yum install -y sysstat //安装命令包

sar -n DEV 1 1 //每1秒显示总共显示1次

rxpck:接收到的数据包个数 txpck:发送的数据包个数，几千正常，上万了可能就被攻击了

rxKB/s:接收数据包的大小 txKB:发送数据包的大小



sar -n DEV -f /var/log/sa/sa01 //查看历史网卡数据

sar -q //查看负载

sar -b //查看磁盘

**nload命令**

1、安装

yum install -y epel-release

yum install -y nload

nload

Curr: 944.00 Bit/s //当前流量

Avg: 1.12 kBit/s //平均流量

Min: 936.00 Bit/s //最小流量

Max: 1.85 kBit/s //最大流量

Ttl: 7.25 MByte //总流量

**监控io性能**

iostat -x //关注 %util，不能太大

iotop //查看哪个进程的io最大

**free命令**

磁盘的数据 ---> 内存（cache｛缓存｝）--> cpu

cpu的数据 ---> 内存 （buff｛缓冲｝）--> 磁盘

total=userd+free+buff/cache

free -h：关注available

**ps命令**

1、ps aux ps -elf //查看运行的进程

VSZ:虚拟内存 RSS:物理内存

STAT：

D:不能中断的进程；R：run状态的进程（一定时间段类使用cpu进程）

S:sleep状态进程 T:暂停的进程；+：前台进程；Z僵尸进程

s:主进程 ；<:高优先级进程；N：低优先级内存；L：内存中被锁定了内存分页

l：多线程进程

**查看网络状态**

netstat -an //查看系统的网络连接状况

netstat -lnp //查看监听端口

netstat -lnpt //查看tcp连接

netstat -an | awk '/^tcp/ {++sta[$NF]} END {for(key in sta) print key,"\t",sta[key]}' //查看所有状态的连接数

yum install -y net-tools //安装ifconfig命令

增加虚拟网卡

/etc/sysconfig/network-scripts //进入该目录

cp ifcfg-eno16777736 ifcfg-eno16777736\:0 //拷贝网卡

vim ifcfg-eno16777736:0

NAME= ifcfg-eno16777736:0

DEVICE=eno16777736:0

BOOTPROTO=static //DHCP测试失败了

IPADDR=192.168.2.115

NETMASK=255.255.255.0

GATEWAY=192.168.2.1

查看网卡是否连接

mii-tool eno16777736



ethtool lo

修改主机名字

hostnamectl set-hostname jz-001

bash

/etc/hostname //存放路径

/etc/resolv.conf //DNS存放路径

**Linux下抓包**

tcpdump -nn -i eno16777736

tcpdump -nn -i eno16777736 not port 22 and host 192.168.1.1 //不要22端口的包，只要IP为192.168.1.1的包

tcpdump -nn -i eno16777736 not port 22 and host 192.168.1.1 -c 100 -w /tmp/1.cap //抓100个包

**firewalld和netfilter**

1、vim /etc/selinux//config //关闭selinux



2、centos7以前的叫netfilter 从7开始叫firewalld

3、systemctl disable firewalld //禁止开机启动

4、systemctl stop firewalld //停止服务

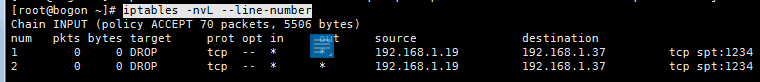
5、yum install -y iptables-services //安装netfilter的包

6、systemctl enable iptables //开启动

7、systemctl start iptables //启动服务

**iptables语法**

1、iptables -nvL --line-number //在规则前面显示行号



2、iptables -D INPUT 1 //根据行号删除规则

3、/usr/local/sbin/iptables.sh

#! /bin/bash

ipt="/usr/sbin/iptables"

$ipt -F

$ipt -P INPUT DROP //修改INPUT的默认规则

$ipt -P OUTPUT ACCEPT

$ipt -P FORWARD ACCEPT

$ipt -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

$ipt -A INPUT -s 192.168.1.0/24 -p tcp --dport 22 -j ACCEPT //指定允许的IP访问22端口

$ipt -A INPUT -p tcp --dport 80 -j ACCEPT //开放80端口

$ipt -A INPUT -p tcp --dport 21 -j ACCEPT //开放21端口

4、iptables -A INPUT -p icmp --icmp-type 8 -j DROP //不允许别人ping自己

**iptables nat表应用**

需求1，有A和B 2台服务器，A服务器有2块网卡，eno16777736为外外往网卡，ens37为内网网卡（192.168.100.1），B服务器只有1块内往网卡（192.168.100.2）,A和B的内网是可以通信的，现在要让B能够上外网

1、在A设备上打开端口转发

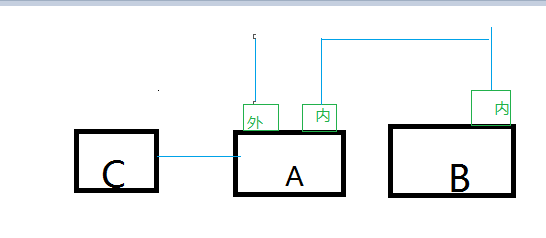
echo "1">/proc/sys/net/ipv4/ip\_forward

2、A上执行

iptables -t nat -A POSTROUTING -s 192.168.100.0/24 -o eno16777736 -j MASQUERADE

###############################

需求2：设备C只能和A通信，但是现在设备C想访问B的22端口，A和B的情况同需求1



备注：C只能够和A通信，而A能够和B通信，这时候需要将B(192.168.100.2)的22端口，映射给A(192.168.1.17：自定义端口)｛这个IP是和C通信的IP｝，当C访问A（192.168.1.17：自定义端口），实际上就是访问的B:22

1、在A设备上打开端口转发

echo "1">/proc/sys/net/ipv4/ip\_forward

2、在A上做端口映射

iptables -t nat -A PREROUTING -d 192.168.1.17 -p tcp --dport 1122 -j DNAT --to 192.168.100.2:22

//这个是进来的包；将192.168.1.17：1122 映射 192.168.100.2：22

iptables -t nat -A POSTROUTING -s 192.168.100.2 -j SNAT --to 192.168.1.17

//这个是出去的包

3、B上设置网关为192.168.100.1 //这个网关就是A的内网IP

**iptables规则备份和恢复**

1、iptables-save > /tmp/ipt.txt //重定向保存规则

2、iptables-restore < /tmp/ipt.txt //恢复规则

3、保存和备份iptables规则

service iptables save //会把规则保存到/etc/sysconfig/iptables

**firewalld的9个zone**

1、systemctl enable firewalld //开机启动

2、systemctl start firewalld //启动服务

3、firewall-cmd --get-zones //查看9个zone



4、firewall-cmd --get-default-zone //查看默认的zone



5、9个zone的介绍

drop：（丢弃），任何接受的网络数据包都被丢弃，没有任何回复。仅能有发送出去的网络连接。

block：（限制）拒绝所有外部连接，允许内部发起的连接，任何接受的网络连接都被IPV4的icmp-host-prohibited信息和ipv6的icmp6-adm-prohibited信息所拒绝。

public：（公共）在公共区域内使用，不能相信网络内的其他计算机不会对你的计算造成危害，只能接受经过选取的连接。

external：（外部）特别是为路由器启用了伪装功能的外部网。你不能信任来自网络的其他计算，不嫩更相信他们不会对你的计算机造成危害，只能接受经过选择的连接。

dmz：（非军事区）用于你的非军事区内的电脑，此区域内可公开访问，可以有限地进入你的内部网络，仅仅接受经过选择的连接。

work：用于工作区。你可以基本相信网络内的其他电脑不会危害你的电脑。仅仅接受经过选择的连接。

home：用于家庭网络。你可以基本相信网络内的其他计算机不会危害你的计算机。仅仅接受经过选择的连接。

internal：用于内部网络，你可以基本信任网络内的其他计算机不会威胁你的计算机，仅仅接受经过选择的连接。

trusted：可接受所有的网络连接

**firewalld关于zone的操作**

1、firewall-cmd --set-default-zone=work //设置默认的zone

2、firewall-cmd --zone=dmz --add-interface=lo //给指定网卡设置zone

3、firewall-cmd --zone=trusted --change-interface=lo //更改指定网卡的zone

4、firewall-cmd --zone=trusted --remove-interface=lo //删除指定网卡的zone

5、firewall-cmd --get-active-zones //查看系统所有网卡所在的区域

**firewalld关于service的操作**

1、firewall-cmd --get-services //查看所有的services

2、firewall-cmd --list-services //查看当前zone上有哪些services

3、firewall-cmd --zone=home --list-services //查看指定zone有哪些services

4、firewall-cmd --zone=work --add-service=http --add-service=ftp //给指定zone增加service

5、firewall-cmd --zone=work --add-service=http --add-service=ftp --permanent //增加service的同时保存到配置文件

6、/etc/firewalld/zones/ //配置文件路径

7、需求：ftp服务自定义端口1121，需要在work zone下面放行ftp

cp /usr/lib/firewalld/services/ftp.xml /etc/firewalld/services/ //靠配配置文件

vim ftp.xml //将端口改成1121

cp /usr/lib/firewalld/zones/work.xml /etc/firewalld/zones/ //默认这个路径下是没有这个work.xml文件的，如果已经存在了就不要拷贝了

vim work.xml

<service name="ftp"/>

firewall-cmd --reload //重新加载服务

**linux任务计划cron**

1、crontab -e //进入编辑任务计划

2、\* \* \* \* \* //分 时 日 月 周

3、0 3 1-10 \*/2 2,5 xxx xxx >>/tmp/123.log 2>>/tmp/123.log //每隔2个月的1到10号，周2和周5的凌晨3点执行任务,后面跟的是日志和错误日志的输出位置

4、systemctl start crond //启动服务

5、cd /var/spool/cron/ //任务计划保存在这个路径下，以创建任务计划的用户为命名