第4天-路由/日志管理

一、静态路由和动态路由

路由器在转发数据时，需要现在路由表中查找相应的路由，有三种途径

> *（1）直连路由：路由器自动添加和自己直连的路由*
>
> *（2）静态路由：管理员手动添加的路由*
>
> *（3）动态路由：由路由协议动态建立的路由*

静态路由

> *缺点 不能动态的反映网络拓扑，当网络发生变化的时候，管理员必须手动的改变路由*
>
> *优点 不会占用路由器太多的cpu和RAM资源，也不会占用太多带宽。*
>
> *如果出于安全的考虑想隐藏网咯的某些部分或者管理员想控制数据转发路径也可以使用静态路由，小网络也可以配置静态路由 因为便捷*

默认路由

> *(1) 实际上默认路由是一种特殊的静态路由，指的是当 路由表中与包的目的地址之间没有匹配的表项时，路由器能够做出选择。如果没有默认路由，那么目的地址在路由表中没有匹配表项的包将被丢弃。*
>
> *(2)默认路由（Default route），如果IP数据包中的目的地址找不到存在的其它路由时，路由器会默认的选择的路由。*
>
> *默认路由为0.0.0.0*
>
> *匹配IP地址时，0表示wildcard，任何值都是可以的，所有0.0.0.0.和任何目的地址匹配都会成功，造成默认路由要求的效果。就是说0 可以匹配任何的IP地址。*

动态路由

> *动态路由是与静态路由相对的一个概念，指路由器能够根据路由器之间的交换的特定路由信息自动地建立自己的路由表，并且能够根据链路和节点的变化适时地进行自动调整。当网络中节点或节点间的链路发生故障，或存在其它可用路由时，动态路由可以自行选择最佳的可用路由并继续转发报文。*

二、Linux路由操作

```
ip方式(rhel7)
查看路由表
[root@qfedu.com ~]#  ip r
default via 10.18.44.1 dev enp0s25
10.18.40.100 via 10.18.44.1 dev enp0s25  proto dhcp  metric 100
10.18.44.0/24 dev enp0s25  proto kernel  scope link  src 10.18.44.196  metric
100
10.18.45.0/24 via 10.18.44.1 dev enp0s25
192.168.122.0/24 dev virbr0  proto kernel  scope link  src 192.168.122.1

删除默认网关
[root@qfedu.com ~]#  ip r d default

删除静态路由：
[root@qfedu.com ~]#  ip r del 10.18.45.0/24

添加默认网关：
[root@qfedu.com ~]#  ip r add default via 10.18.44.1 dev enp0s25

添加静态路由：
[root@qfedu.com ~]#  ip r add 10.18.45.0/24 via 10.18.44.1 dev enp0s25
```

三、实战

1、添加默认路由

2、Linux服务器配置静态路由并测试

##四、端口和服务解析

1、网络端口

> *在网络技术中，端口（Port）有好几种意思。集线器、交换机、路由器的端口指的是连接其他网络设备的接口，如RJ-45端口、Serial端口等。我们这里所指的端口不是指物理意义上的端口，而是特指TCP/IP协议中的端口，是逻辑意义上的端口。*
>
> *如果把IP地址比作一间房子，端口就是出入这间房子的门。真正的房子只有几个门，但是一个IP地址的端口可以有65536（即：2^16）个之多！端口是通过端口号来标记的，端口号只有整数，范围是从0 到65535（2^16-1）。*
>
> *在Internet上，各主机间通过TCP/IP协议发送和接收数据包，各个数据包根据其目的主机的ip地址来进行互联网络中的路由选择,把数据包顺利的传送到目的主机。大多数操作系统都支持多程序（进程）同时运行，那么目的主机应该把接收到的数据包传送给众多同时运行的进程*

> *中的哪一个呢？显然这个问题有待解决，端口机制便由此被引入进来。*
>
> *本地操作系统会给那些有需求的进程分配协议端口（protocol port，即我们常说的端口），每个协议端口由一个正整数标识，如：80，139，445，等等。当目的主机接收到数据包后，将根据报文首部的目的端口号，把数据发送到相应端口，而与此端口相对应的那个进程将会领取数据并等待下一组数据的到来。说到这里，端口的概念似乎仍然抽象，那么继续跟我来，别走开。*
>
> *端口其实就是队，操作系统为各个进程分配了不同的队，数据包按照目的端口被推入相应的队中，等待被进程取用，在极特殊的情况下，这个队也是有可能溢出的，不过操作系统允许各进程指定和调整自己的队的大小。*
>
> *不光接受数据包的进程需要开启它自己的端口，发送数据包的进程也需要开启端口，这样，数据包中将会标识有源端口，以便接受方能顺利地回传数据包到这个端口。*

2、常见服务管理操作

```
[root@qfedu.com ~]# systemctl list-units
[root@qfedu.com ~]# systemctl list-unit-files
[root@qfedu.com ~]# systemctl start  vsftpd
[root@qfedu.com ~]# systemctl status  vsftpd
[root@qfedu.com ~]# systemctl status  vsftpd -l
[root@qfedu.com ~]# systemctl stop vsftpd
[root@qfedu.com ~]# systemctl restart vsftpd
[root@qfedu.com ~]# systemctl reload  vsftpd
[root@qfedu.com ~]# systemctl enable vsftpd
[root@qfedu.com ~]# systemctl disable vsftpd
```

五、日志

1、日志重要性

> *Linux系统日志对管理员来说，是了解系统运行的主要途径，因此需要对Linux 日志系统有个详细的了解。*
>
> *Linux 系统内核和许多程序会产生各种错误信息、告警信息和其他的提示信息，这些各种信息都应该记录到日志文件中，完成这个过程的程序就是rsyslog，rsyslog 可以根据日志的类别和优先级将日志保存到不同的文件中。*

2、常见系统日志

/var/log/message：记录Linux操作系统常见的系统和服务错误信息
/var/log/boot.log：录了系统在引导过程中发生的事件，就是Linux系统开机自检过程显示的信息
/var/log/lastlog ：记录最后一次用户成功登陆的时间、登陆IP等信息（一般通过命令 lastlog 查看）
/var/log/secure ：Linux系统安全日志，记录用户和工作组变坏情况、用户登陆认证情况
/var/log/btmp ：记录Linux登陆失败的用户、时间以及远程IP地址
/var/log/wtmp：该日志文件永久记录每个用户登录、注销及系统的启动、停机的事件，使用last命令查看

3、rsyslog日志管理

3.1、日志类型

auth pam 产生的日志

authpriv ssh，ftp 等登录信息的验证信息

cron 时间任务相关

kern 内核

lpr 打印

mail 邮件

mark(syslog)-rsyslog 服务内部的信息，时间标识

news 新闻组

user 用户程序产生的相关信息

3.2、日志优先级

日志级别分为：7种日志级别代号0-7

0 debug 有调试信息的，日志信息最多

1 info 一般信息的日志，最常用

2 notice 最具有重要性的普通条件的信息

3 warning 警告级别

4 err 错误级别，阻止某个功能或者模块不能正常工作的信息

5 crit 严重级别，阻止整个系统或者整个软件不能工作的信息

6 alert 需要立刻修改的信息

7 emerg 内核崩溃等严重信息

none 什么都不记录

3.3、自定义日志

```
[root@qfedu.com ~]#vim /etc/rsyslog.conf
日志对象(设备):你要对什么东东做日志
日志级别:级别越低，信息越多
日志文件:存储日志的文件

日志对象.日志级别  日志文件
. 大于或者等于后面指定的日志级别
.= 等于后面指定的日志级别
.! 非

例：
*.*                     /var/log/mylog
kern.err                /var/log/kernel.log
*.info;mail.none    /var/log/big.log
mail.info               /var/log/mail.log
cron.info;cron.!err /var/log/newcron
cron.info               /var/log/newcron

重启日志服务：
[root@qfedu.com ~]#systemctl restart rsyslog
```

4、logrotate日志轮转

4.1、日志轮转

```
[root@qfedu.com ~]# vim /etc/logrotate.conf
  //全局配置
  weekly    轮转周期   默认一周轮转一次
  rotate 4   轮转次数   默认轮转4次
  create     创建新文件
  dateext 以轮转时刻的时间作为轮转文件的结尾
  //局部配置
  include /etc/logrotate.d
  missingok  在文件不存在的时候也不报错
  create 0644 root utmp
```

```
强制轮转：
[root@qfedu.com ~]#logrotate -s /var/lib/logrotate/logrotate.status
/etc/logrotate.conf
-s 指定最后的日志轮转记录文件为/var/lib/logrotate/logrotate.status
```

4.2、日志轮转实例

```
1：测试日志轮转，轮转文件/var/log/yum.log
[root@qfedu.com ~]# vim /etc/logrotate.d/yum
/var/log/yum.log {
    missingok
#   notifempty
#   size 30k
#   yearly
    daily
    rotate 3
    create 0777 root root
}

测试：
[root@qfedu.com ~]# logrotate /etc/logrotate.conf    //手动轮转
[root@qfedu.com ~]# ls /var/log/yum*
/var/log/yum.log  /var/log/yum.log-20170331
[root@qfedu.com ~]# grep 'yum' /var/lib/logrotate/logrotate.status      //记录
所有日志文件最近轮转的时间
"/var/log/yum.log" 2017-3-31-10:0:23

[root@qfedu.com ~]# date 09011000
[root@qfedu.com ~]# logrotate -s /var/lib/logrotate/logrotate.status
/etc/logrotate.conf

2：日志安全，操作日志的隐藏权限
/etc/logrotate.d/messages
建议测试时先把/etc/logrotate.d/syslog中messages删除
/var/log/messages {
  prerotate
    chattr -a /var/log/messages
  endscript

  #notifempty
  daily
  create 0600 root root
  missingok
  rotate 5

  postrotate
```

```
        chattr +a /var/log/messages
    endscript
}

3：为多个日志文件配置日志轮转
[root@qfedu.com ~]#vim /etc/logrotate.d/syslog
/var/log/cron
/var/log/maillog
/var/log/secure
/var/log/spooler
{
    missingok
    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
    endscript
}
```

4.3、日志轮转中重启服务的重要性

> *日志轮转配置文件中重启服务的脚本 是为了把新的日志内容写入到新的日志文件里 因为旧的日志文件被轮转只是改了个名字，INODE并没有变，但是日志程序是按日志文件的 inode 号识别文件的，所以需要重启日志以改变日志文件为新的文件*

六、实战

1、使用rsyslog自定义日志

2、使用Logrotate对日志进行日志轮转