# 第11章 安装和配置Web服务器

LAMP Linux Apache MySQL PHP

### Linux上安装Apache服务创建网站

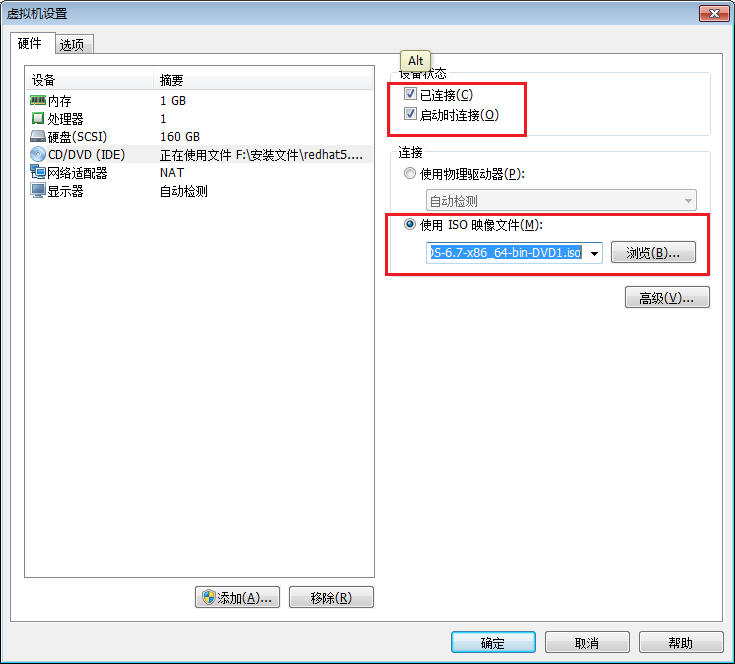
可以使用本地YUM源安装Apache也可以通过Internet的源来安装.

在文件的最后添加如下内容。

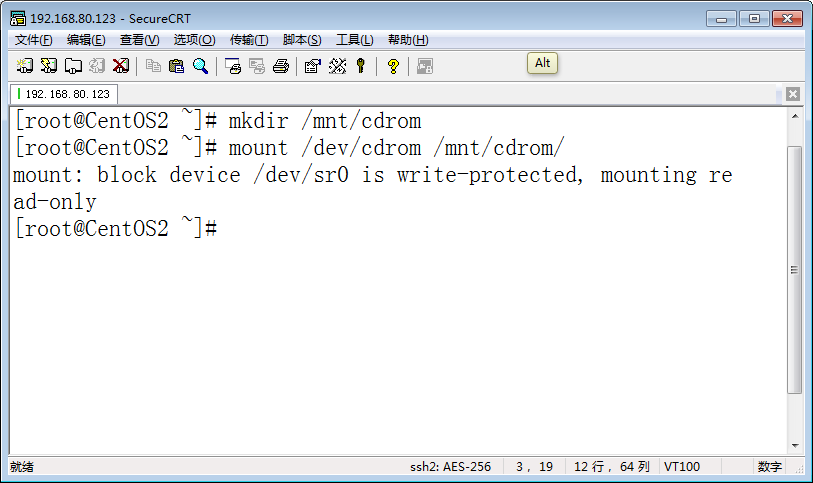
配置本地源

本地yum源

确保虚拟机光驱加载了CentOS的安装光盘



挂载光驱到/mnt/cdrom

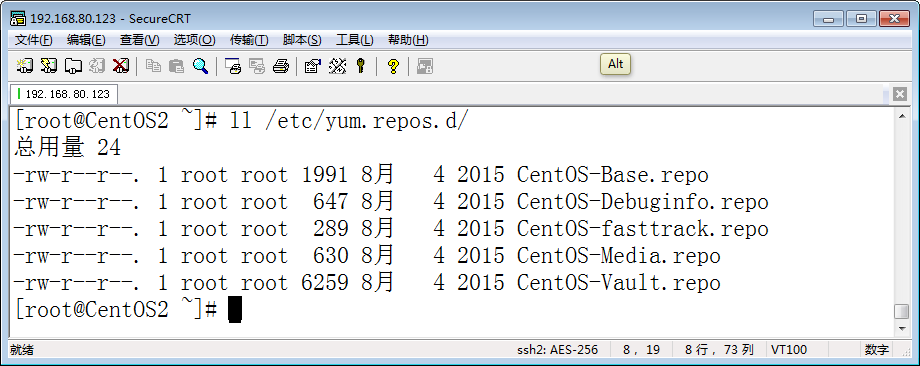


查看yum源配置文件

/etc/yum.repos.d/

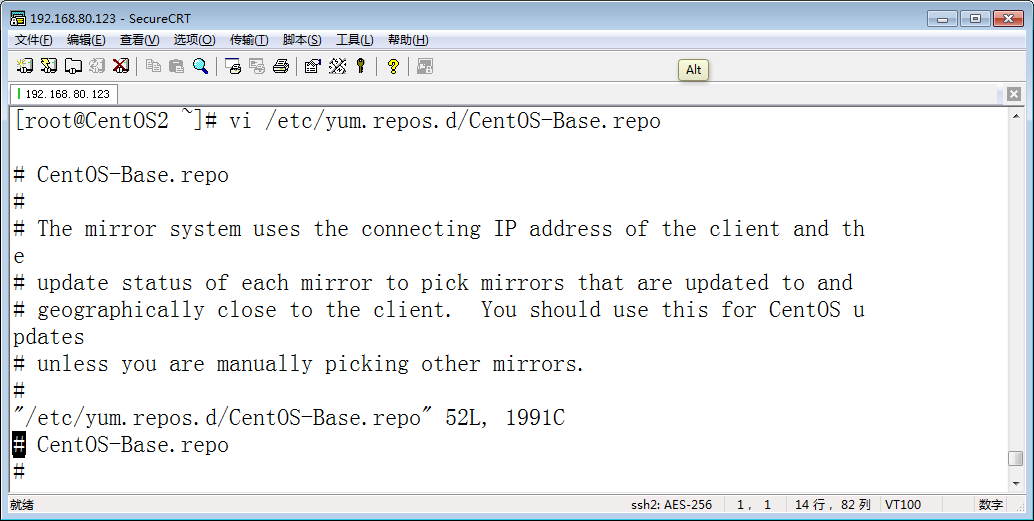
在该目录中所有的\*.repo文件都是一个yum仓库

一个文件可以有多个yum仓库

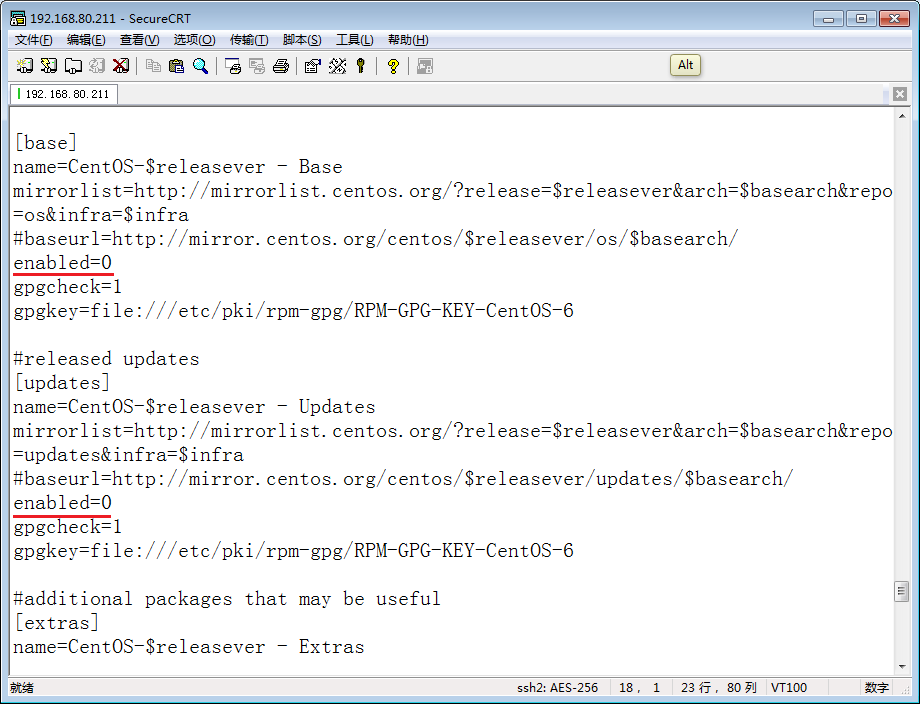


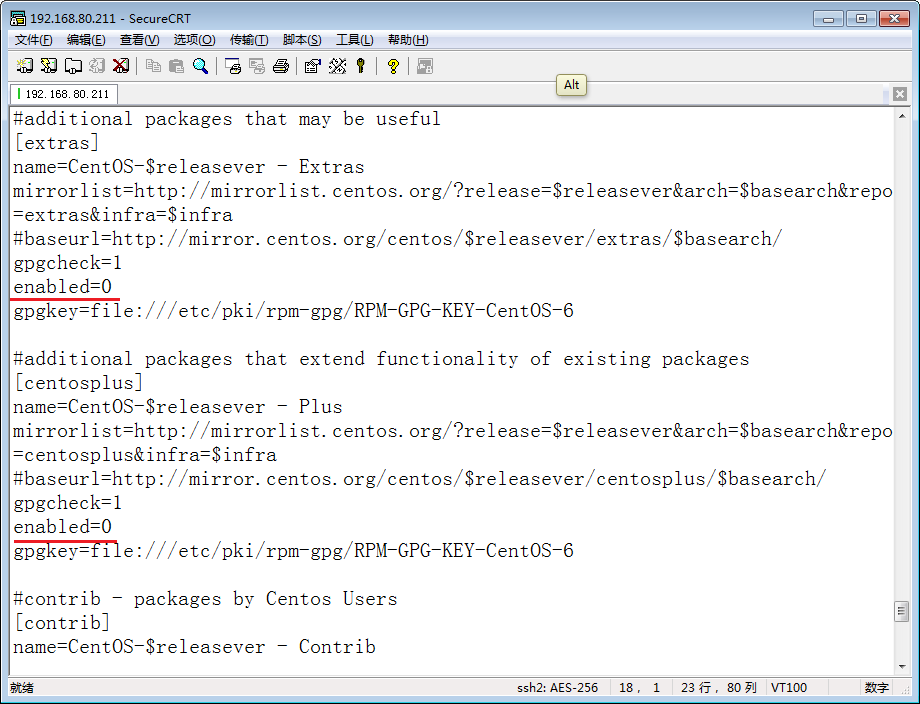
编辑第一个文件，默认该文件指定了Internet上的yum源。

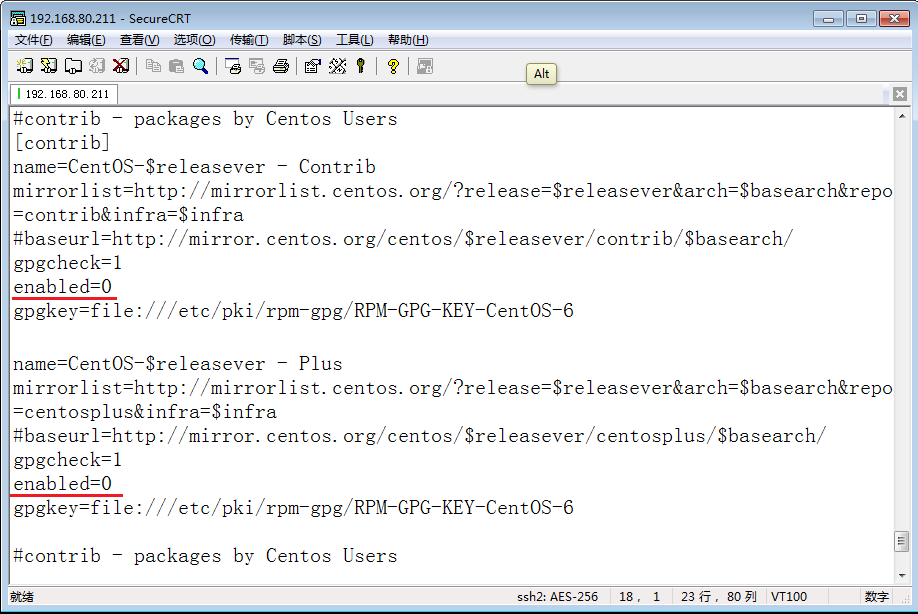
vi /etc/yum.repos.d/CentOS-Base.repo



将各个yum源添加enabled=0 就代表禁用了该yum源







在文件的最后添加如下内容。

配置本地源

[CentOS-local-source]

name=CentOS 6.4

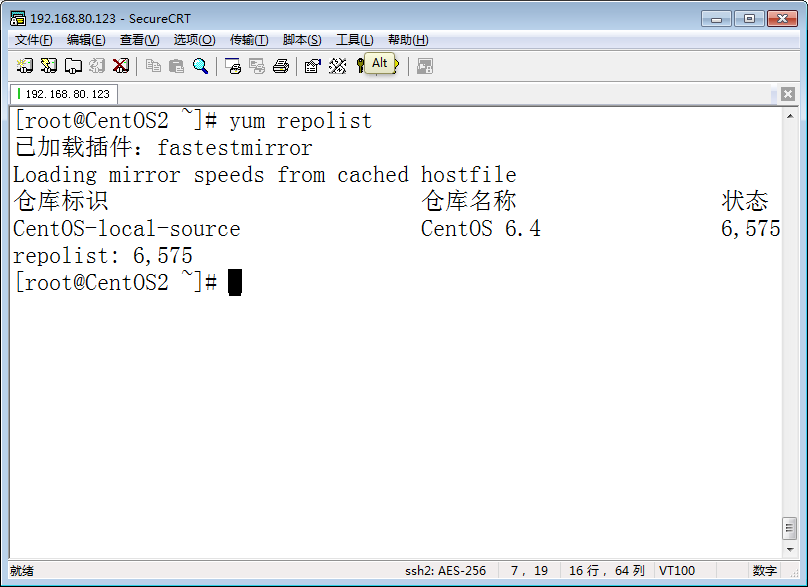
baseurl=file:///mnt/cdrom/

enabled=1

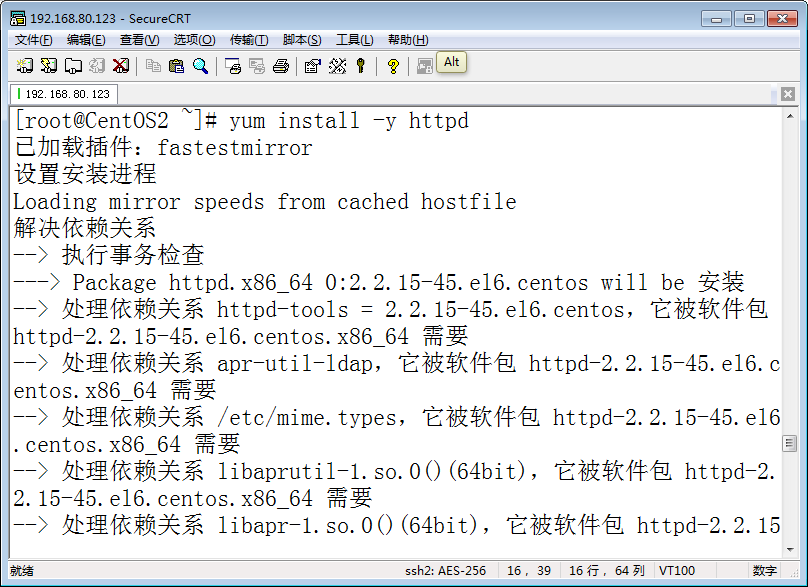
gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6



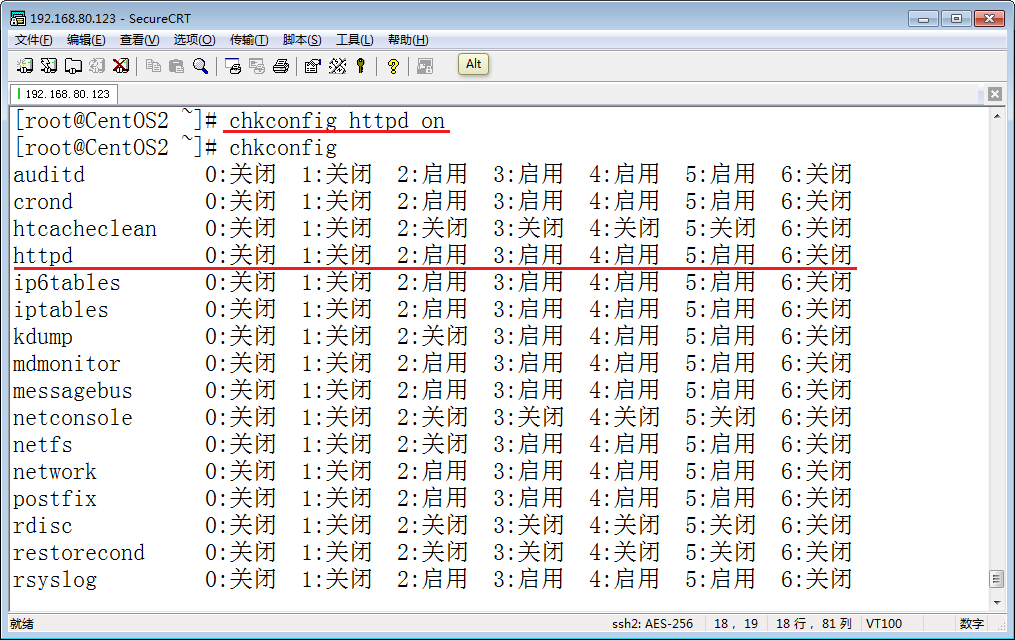
[root@CentOS2 ~]# yum repolist

[root@CentOS2 ~]# yum install -y httpd

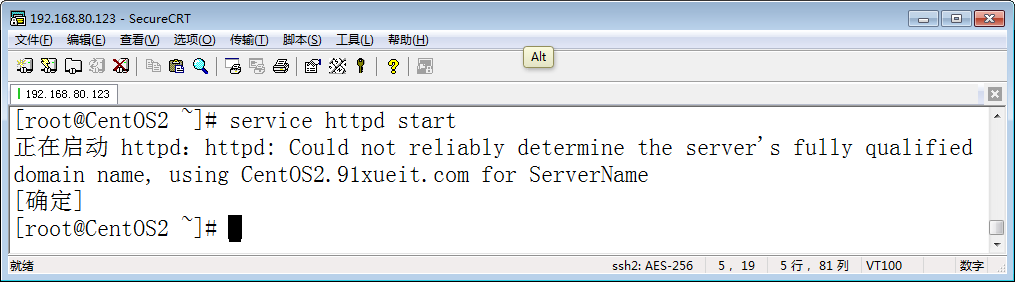


设置httpd服务开机自动启动

[root@CentOS2 ~]# chkconfig httpd on

[root@CentOS2 ~]# chkconfig 

[root@CentOS2 ~]# service httpd start



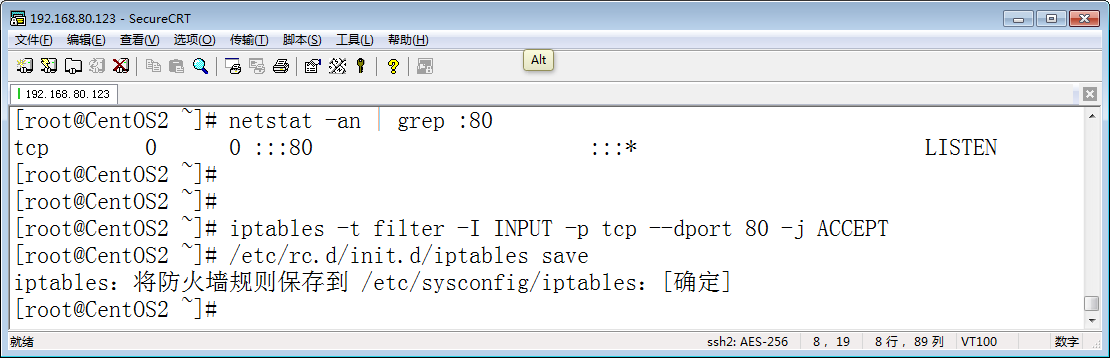
查看侦听的端口

[root@CentOS2 ~]# netstat -an | grep :80

tcp 0 0 :::80 :::\* LISTEN

防火墙打开TCP的80端口

[root@CentOS2 ~]# iptables -t filter -I INPUT -p tcp --dport 80 -j ACCEPT

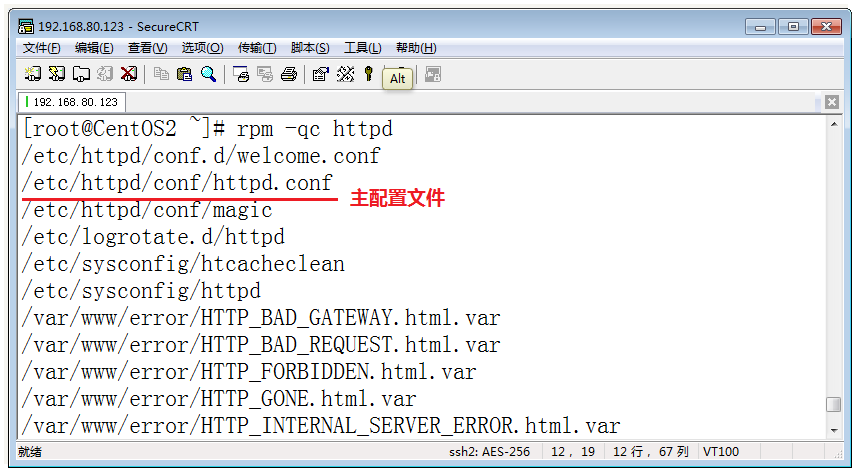
[root@CentOS2 ~]# /etc/rc.d/init.d/iptables save

### Apache配置文件

### 查看Apache版本

查看httpd服务相关的文件

[root@WebServer ~]# rpm -ql httpd

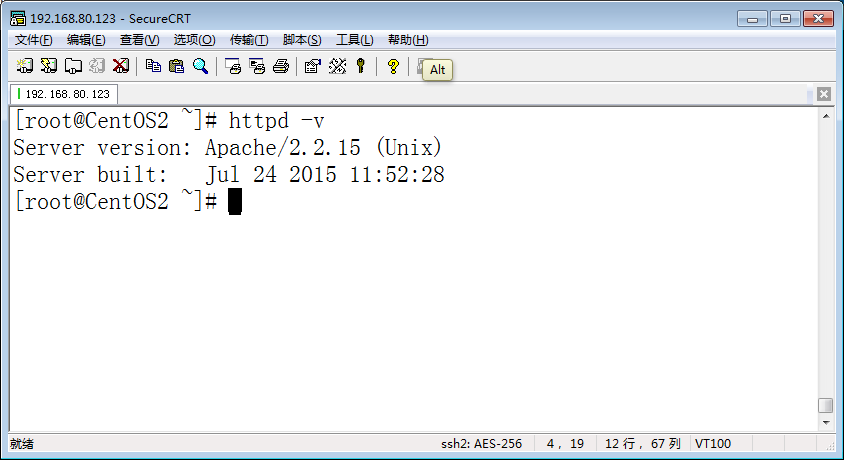


查看Apache版本

[root@CentOS2 ~]# httpd -v

Server version: Apache/2.2.15 (Unix)

Server built: Jul 24 2015 11:52:28



　　1、所需要的软件包：httpd httpd-devel httpd-manual

　　2、端口：80（http） 443（https）

　　3、主配置文件：/etc/httpd/\* 、/var/www/\*

　　4、默认主页目录：/var/www/html

　　5、脚本： /etc/init.d/httpd

　　6、守护进程：/usr/sbin/httpd

查看和httpd服务相关的配置文件，红色部分是主配置文件

[root@WebServer ~]# rpm -qc httpd

/etc/httpd/conf.d/welcome.conf

/etc/httpd/conf/httpd.conf

/etc/httpd/conf/magic

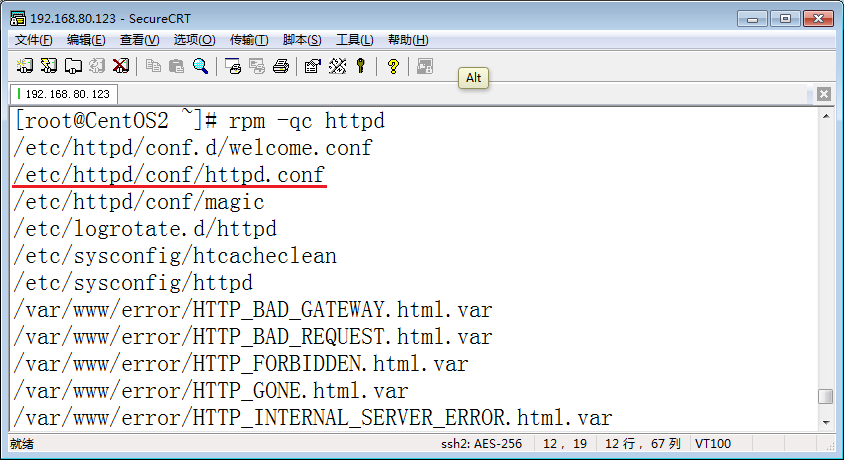
/etc/logrotate.d/httpd

/etc/sysconfig/htcacheclean

/etc/sysconfig/httpd

/var/www/error/HTTP\_BAD\_GATEWAY.html.var

/var/www/error/HTTP\_BAD\_REQUEST.html.var



## Apache主配置文件参数

打开主配置文件

[root@CentOS2 ~]# vi /etc/httpd/conf/httpd.conf

全局配置

### 设置ServerTokens

禁止显示或发送Apache版本号（设置ServerTokens）

默认地，服务器HTTP响应头会包含apache和php版本号。像下面的，这是有危害的，因为这会让黑客通过知道详细的版本号而发起已知该版本的漏洞攻击。

ServerTokens 可以设为列各值中的一个：

Full | OS | Minor | Minimal | Major | Prod

# vi httpd.conf

下面是ServerTokens的一些可能的赋值：

ServerTokens Prod 显示“Server: Apache”

ServerTokens Major 显示 “Server: Apache/2″

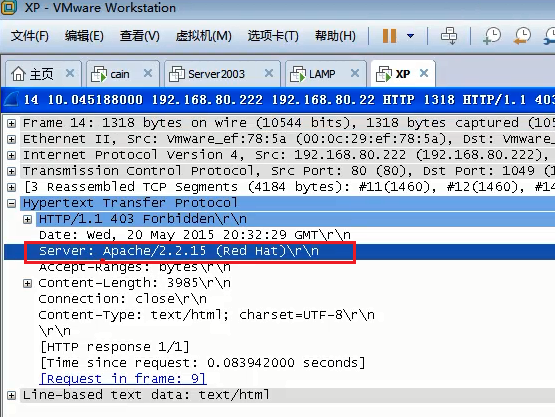
ServerTokens Minor 显示“Server: Apache/2.2″

ServerTokens Min 显示“Server: Apache/2.2.17″

ServerTokens OS 显示 “Server: Apache/2.2.17 (Unix)”

ServerTokens Full 显示 “Server: Apache/2.2.17 (Unix) PHP/5.3.5″ (如果你这指定任何的值，这个是默认的返回信息)

如下图所示，客户端访问时数据包中带着Apache版本信息



### Serverroot

#ServerRoot:指出服务器保存其配置、出错和日志文件等的根目录。#路径的结尾不要添加斜线。

如果配置文件中路径不是/开始 就会加上Serverroot构成路径

ServerRoot "/etc/httpd"

[root@WebServer ~]# ll /etc/httpd/

### MPM优化服务器性能

Apache 2.X 支持插入式并行处理模块，称为多路处理模块（MPM）。在编译apache时必须选择也只能选择一个MPM，对类UNIX系统，有几个不同的MPM可供选择，它们会影响到apache的速度和可伸缩性。

prefork模式

Prefork MPM : 这个多路处理模块(MPM)实现了一个非线程型的、预派生的web服务器，它的工作方式类似于Apache 1.3。它适合于没有线程安全库，需要避免线程兼容性问题的系统。它是要求将每个请求相互独立的情况下最好的MPM，这样若一个请求出现问题就不会影响到其他请求。

这个MPM具有很强的自我调节能力，只需要很少的配置指令调整。最重要的是将MaxClients设置为一个足够大的数值以处理潜在的请求高峰，同时又不能太大，以致需要使用的内存超出物理内存的大小。

Worker 模式

Worker MPM : 此多路处理模块(MPM)使网络服务器支持混合的多线程多进程。由于使用线程来处理请求，所以可以处理海量请求，而系统资源的开销小于基于进程的MPM。但是，它也使用了多进程，每个进程又有多个线程，以获得基于进程的MPM的稳定性。

每个进程可以拥有的线程数量是固定的。服务器会根据负载情况增加或减少进程数量。一个单独的控制进程(父进程)负责子进程的建立。每个子进程可以建立ThreadsPerChild数量的服务线程和一个监听线程，该监听线程监听接入请求并将其传递给服务线程处理和应答。

不管是Worker模式或是Prefork 模式，Apache总是试图保持一些备用的(spare)或者是空闲的子进程（空闲的服务线程池）用于迎接即将到来的请求。这样客户端就不需要在得到服务前等候子进程的产生。

Event模式

Event MPM：以上两种稳定的MPM方式在非常繁忙的服务器应用下都有些不足。尽管HTTP的Keepalive方式能减少TCP连接数量和网络负载，但是 Keepalive需要和服务进程或者线程绑定，这就导致一个繁忙的服务器会耗光所有的线程。 Event MPM是解决这个问题的一种新模型，它把服务进程从连接中分离出来。在服务器处理速度很快，同时具有非常高的点击率时，可用的线程数量就是关键的资源限 制，此时Event MPM方式是最有效的。一个以Worker MPM方式工作的繁忙服务器能够承受每秒好几万次的访问量（例如在大型新闻服务站点的高峰时），而Event MPM可以用来处理更高负载。值得注意的是，Event MPM不能在安全HTTP（HTTPS）访问下工作。

### 增加或更改Web服务使用的端口

配置Apache服务使用的端口，增加81端口

Listen 80

Listen 81

一、修改

/etc/httpd/conf/httpd.conf

文件中的监听端口号

Listen 80

再增加

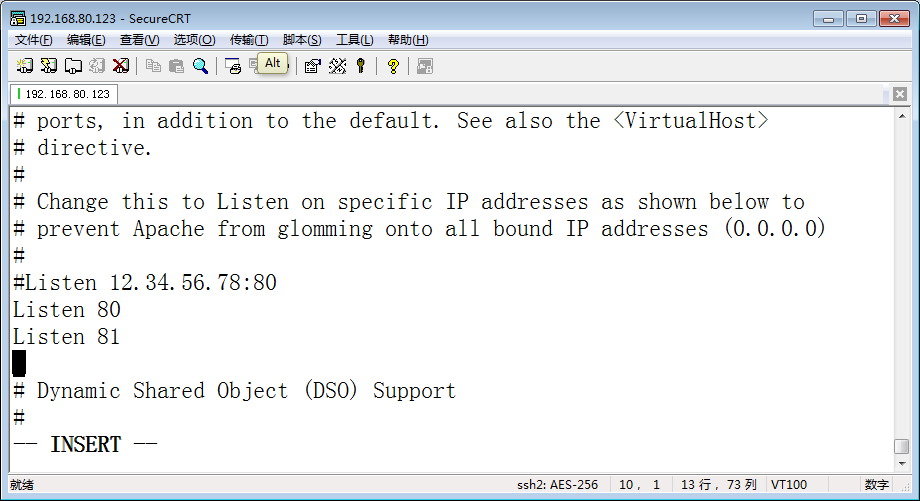
Listen 80

Listen 81

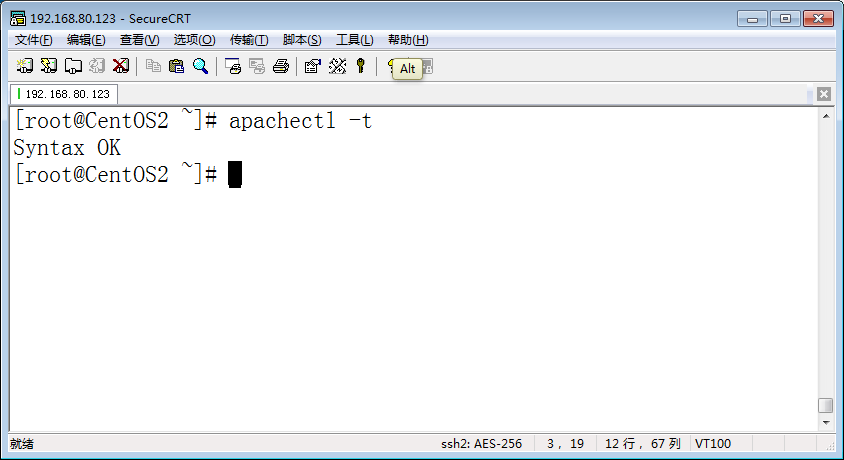
重启服务，查看侦听的端口有了81，当然你需要配置防火墙打开TCP的81端口

[root@CentOS2 ~]# service httpd restart

[root@CentOS2 ~]# netstat -an | grep :81



### 检查配置文件是否有错误



### 关闭SELinux

查看SELinux是否启用

[root@WebServer ~]# sestatus

SELinux status: enabled

SELinuxfs mount: /selinux

Current mode: enforcing

Mode from config file: enforcing

Policy version: 24

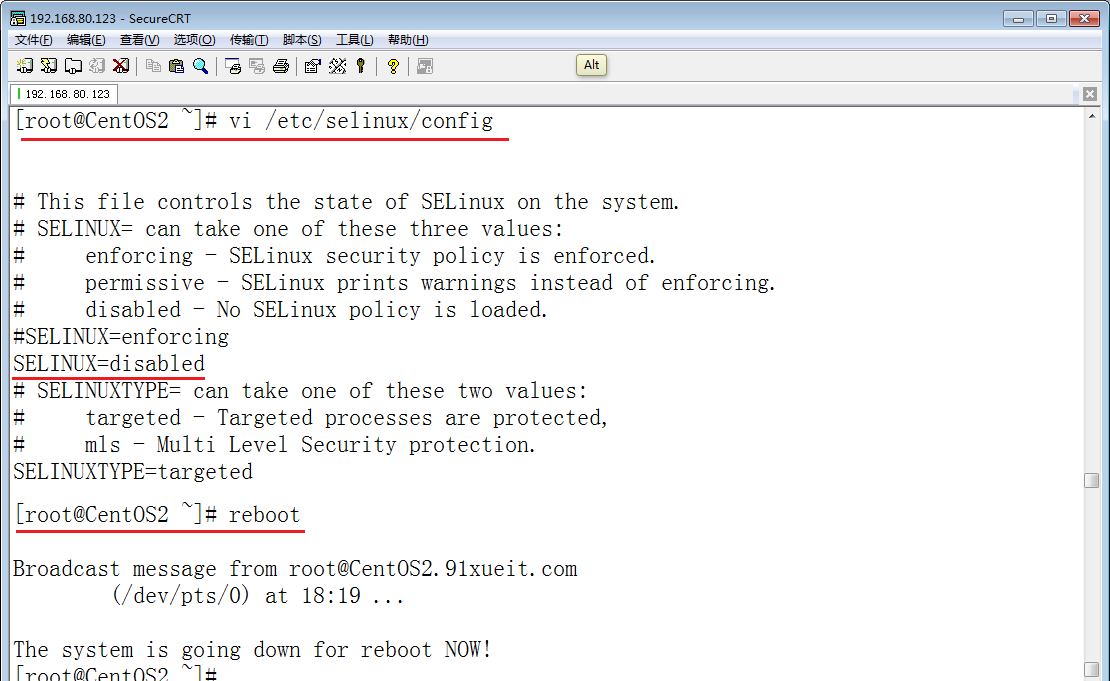
Policy from config file: targeted

更改配置文件，

[root@CentOS2 ~]# vi /etc/selinux/config

#SELINUX=enforcing

SELINUX=disabled

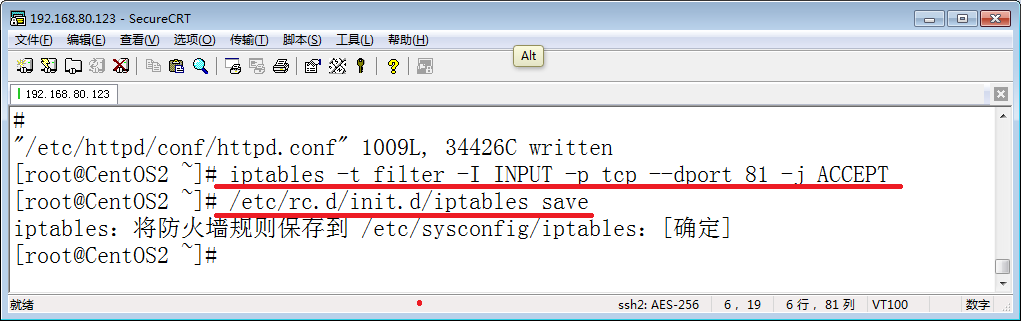


### 在防火墙中开放新添加的端口

[root@WebServer ~]# iptables -t filter -I INPUT -p tcp --dport 81 -j ACCEPT

保存防火墙配置

[root@WebServer ~]# /etc/rc.d/init.d/iptables save



## 创建虚拟机主机

在CentOS上创建两个网站，使用域名来区分。

### 使用WinSCP连接Linux

取消该注释，否则就WinSCPU连接，过一会儿就中断。

更改配置文件，取消红色行前的#

[root@WebServer ~]# vi /etc/ssh/sshd\_config

#LoginGraceTime 2m

PermitRootLogin yes

#StrictModes yes

#MaxAuthTries 6

#MaxSessions 10

重启sshd服务

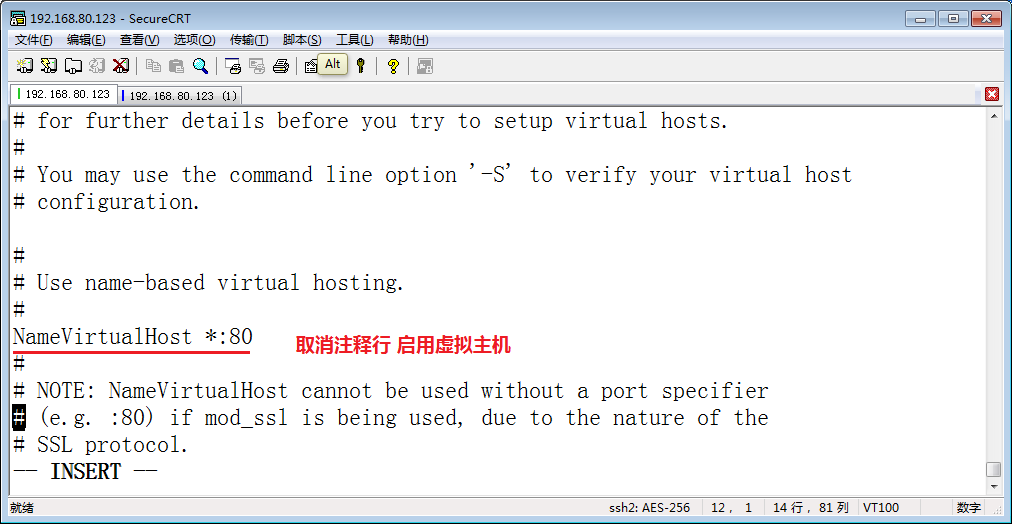
[root@WebServer ~]# service sshd restart



### 配置虚拟机主机

编辑配置文件

NameVirtualHost \*:80



将以下内容拷贝到配置文件最后。

<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host.example.com

DocumentRoot /var/www/sogou

ServerName www.sogou.com

DirectoryIndex sogou.htm

ErrorLog logs/dummy-host.example.com-error\_log

CustomLog logs/dummy-host.example.com-access\_log common

</VirtualHost>

<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host.example.com

DocumentRoot /var/www/baidu

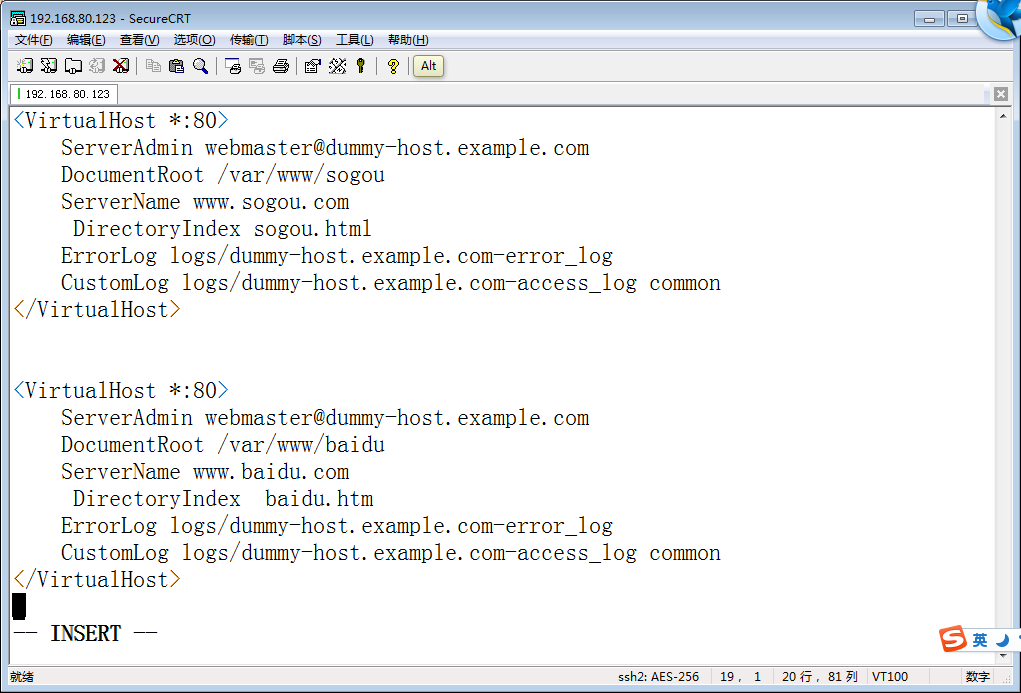
ServerName www.baidu.com

DirectoryIndex baidu.htm

ErrorLog logs/dummy-host.example.com-error\_log

CustomLog logs/dummy-host.example.com-access\_log common

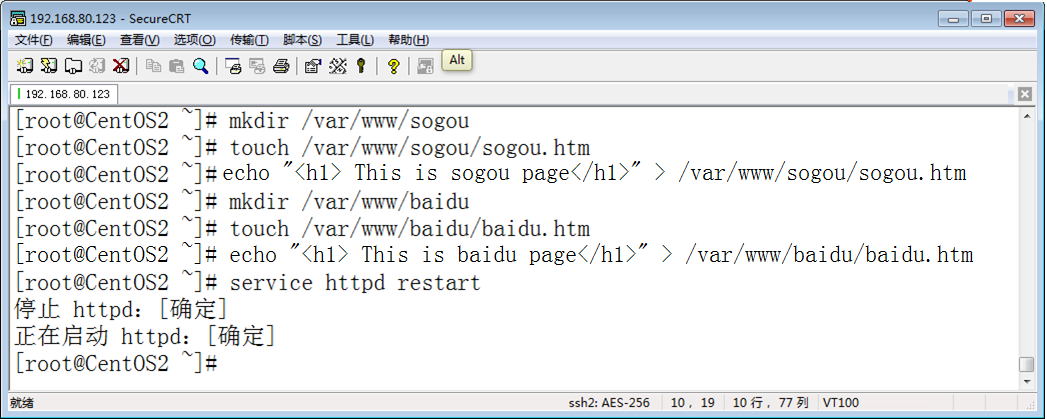
</VirtualHost>



以下命令创建两个网页

[root@CentOS2 ~]# echo "<h1> This is baidu page</h1>" > /var/www/baidu/baidu.htm

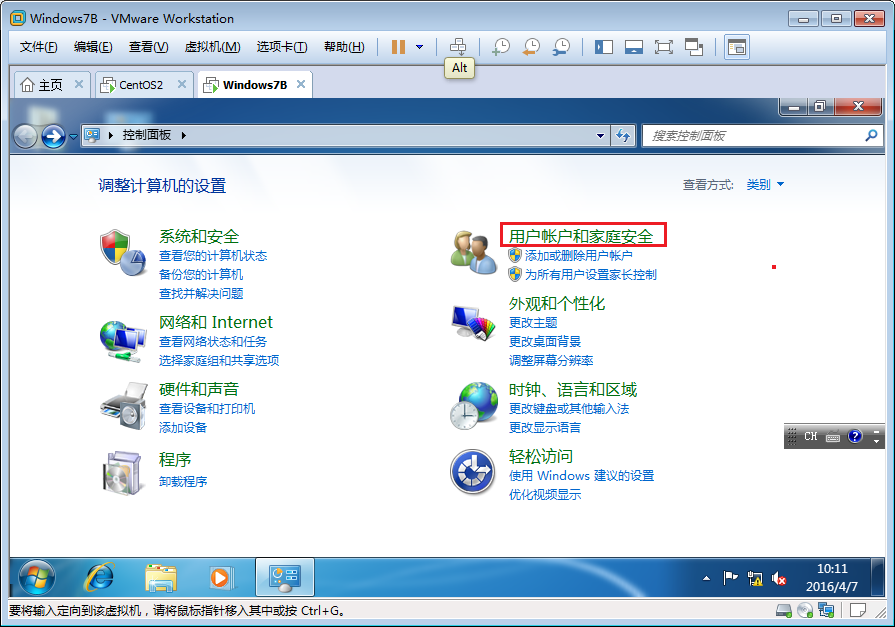
[root@CentOS2 ~]# echo "<h1> This is sogou page</h1>" > /var/www/sogou/sogou.htm



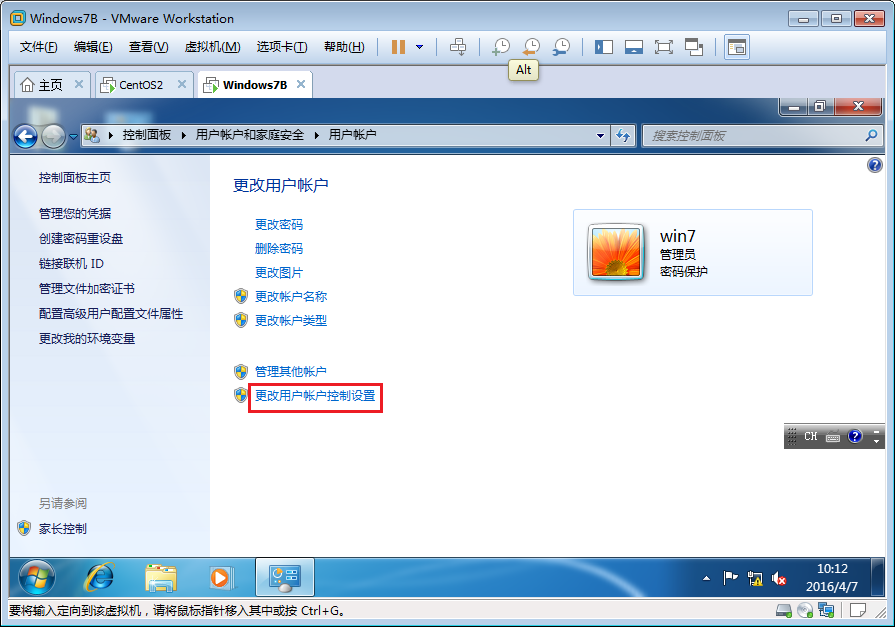
### 通过域名访问这两个网站

先更改Windows7的用户账户控制级别，否则不能够更改hosts文件的内容。

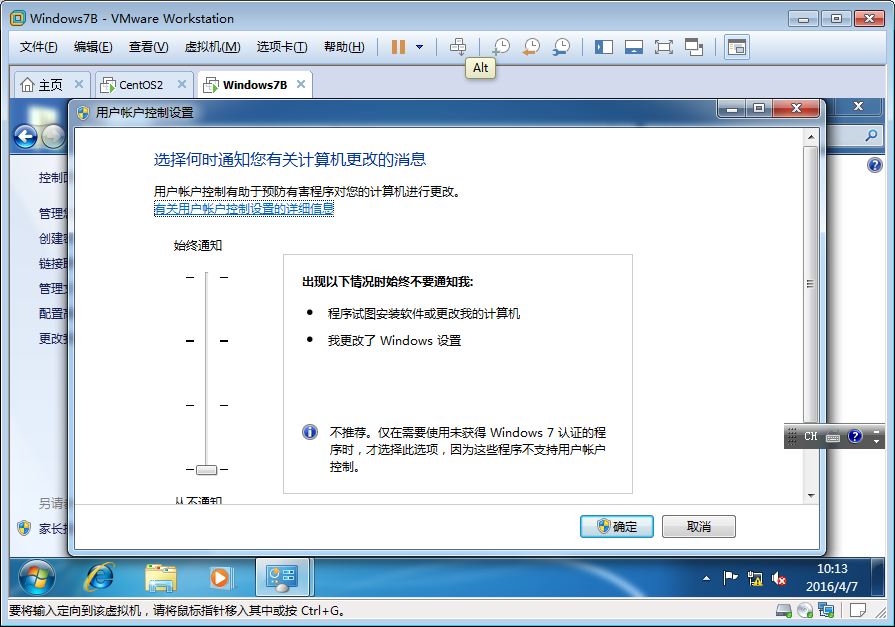
在Windows7B hosts文件添加域名和IP地址的对应关系



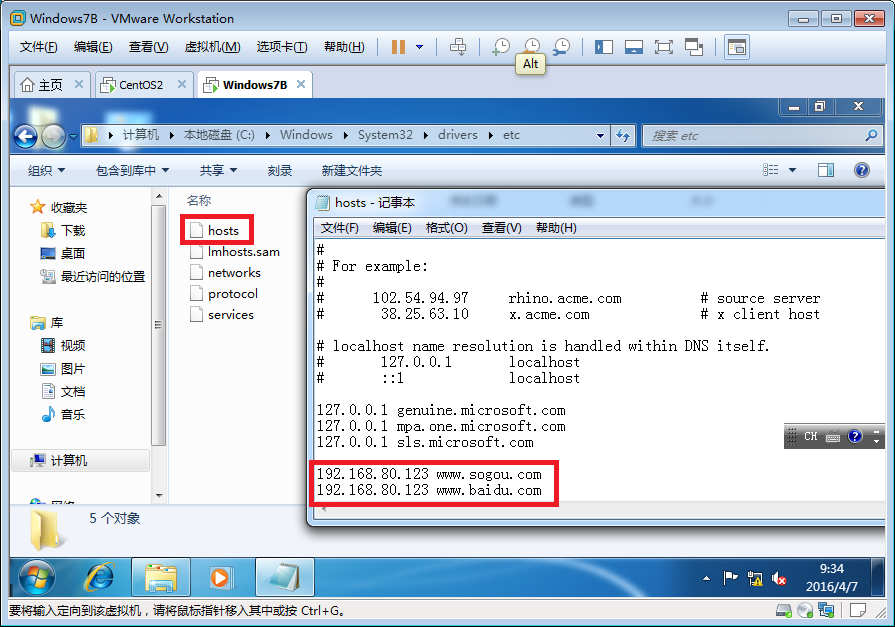




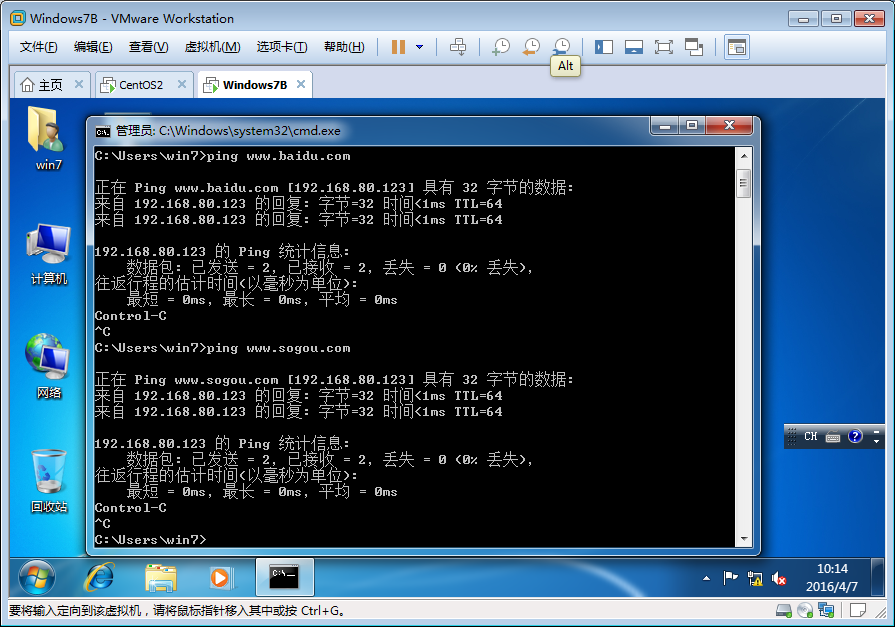
重启系统

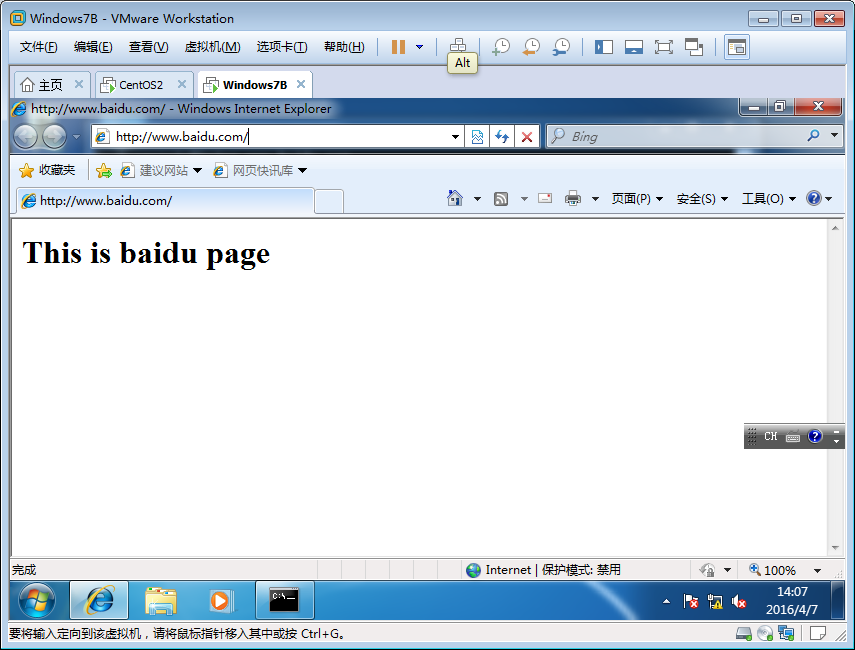


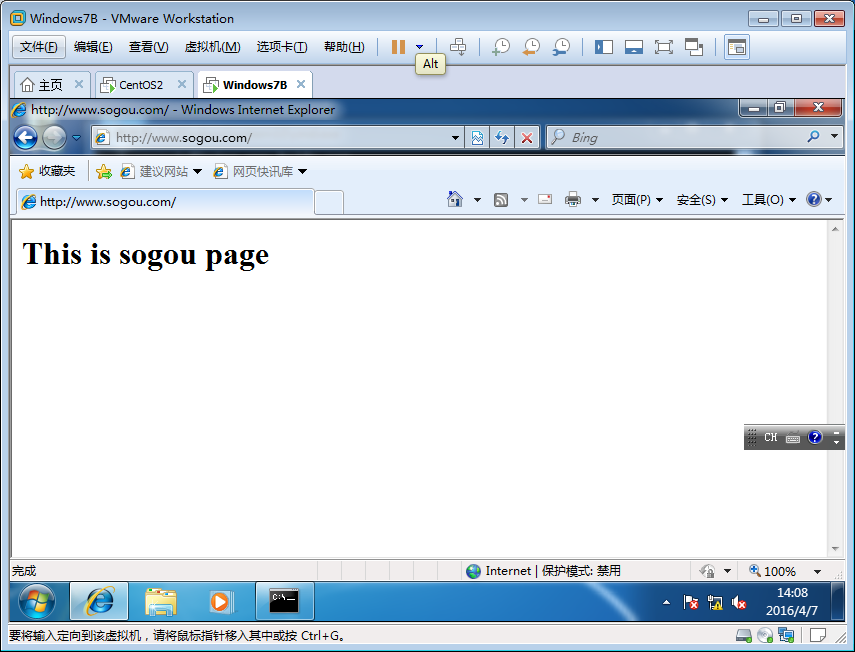
重启后向hosts文件添加内容



可以看到Windows7现在可以将这两个域名解析到Linux的地址







## 创建虚拟目录

创建网站的虚拟目录

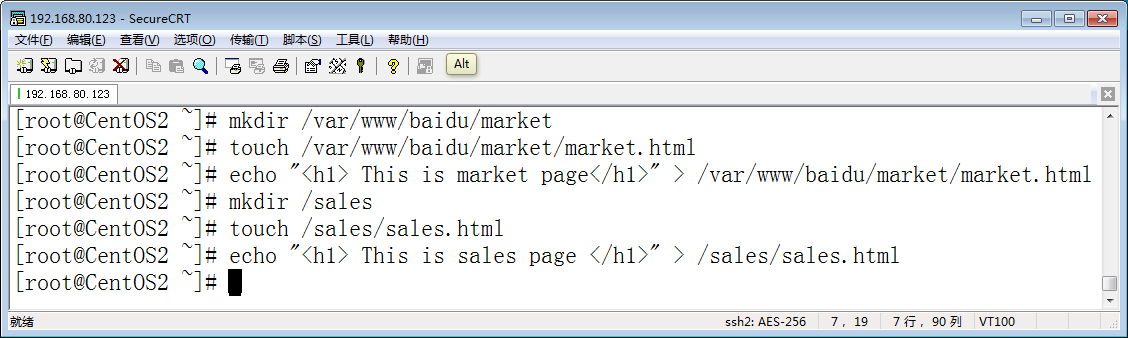
[root@CentOS2 ~]# mkdir /var/www/baidu/market

[root@CentOS2 ~]# touch /var/www/baidu/market/market.html

[root@CentOS2 ~]# echo "<h1> This is market page</h1>" > /var/www/baidu/market/market.html[root@CentOS2 ~]# mkdir /sales

[root@CentOS2 ~]# touch /sales/sales.html

[root@CentOS2 ~]# echo "<h1> This is sales page </h1>" > /sales/sales.html



<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host.example.com

DocumentRoot /var/www/baidu

ServerName www.baidu.com

DirectoryIndex baidu.htm

ErrorLog logs/dummy-host.example.com-error\_log

CustomLog logs/dummy-host.example.com-access\_log common

#在网站文件夹中虚拟目录，定义该虚拟目录的首页

<Directory "/var/www/baidu/market">

DirectoryIndex market.html

</Directory>

#网页在/sales目录中 需要创建别名客户端使用别名来访问http://www.baidu.com/s

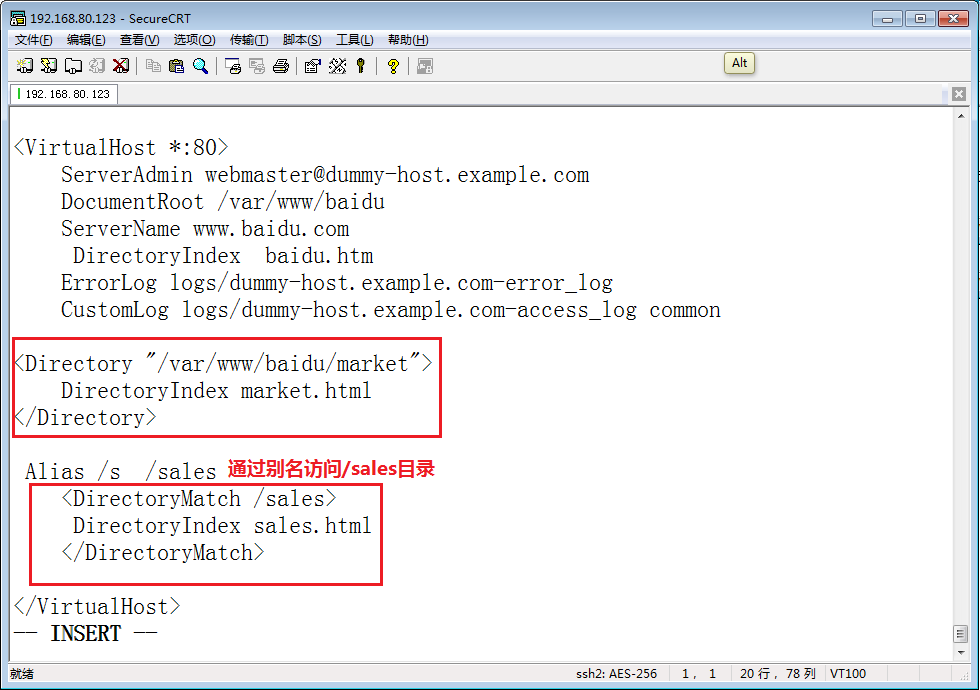
Alias /s /sales

<DirectoryMatch /sales>

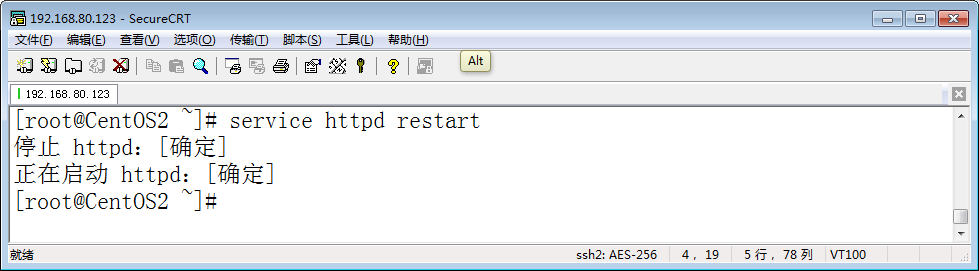
DirectoryIndex sales.html

</DirectoryMatch>

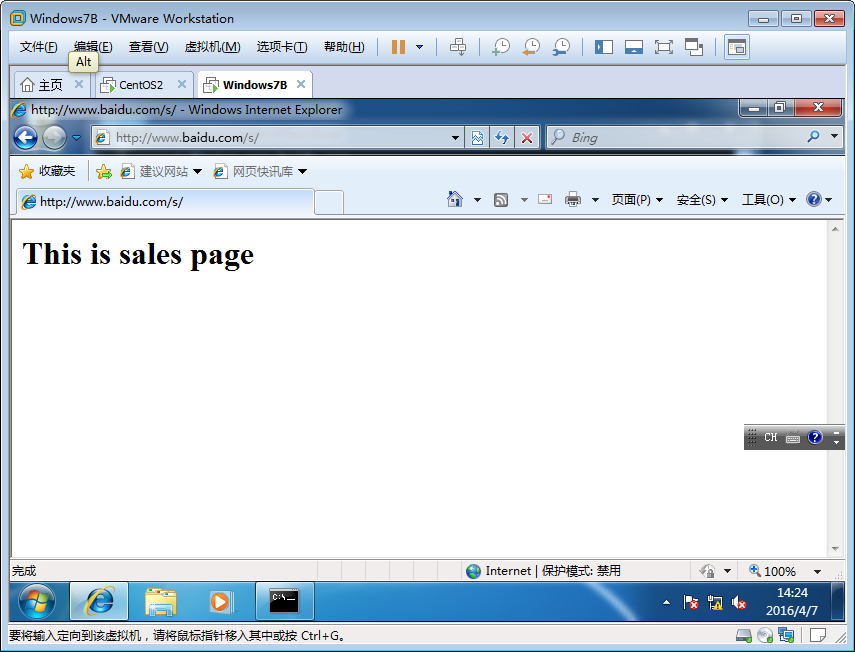
</VirtualHost>



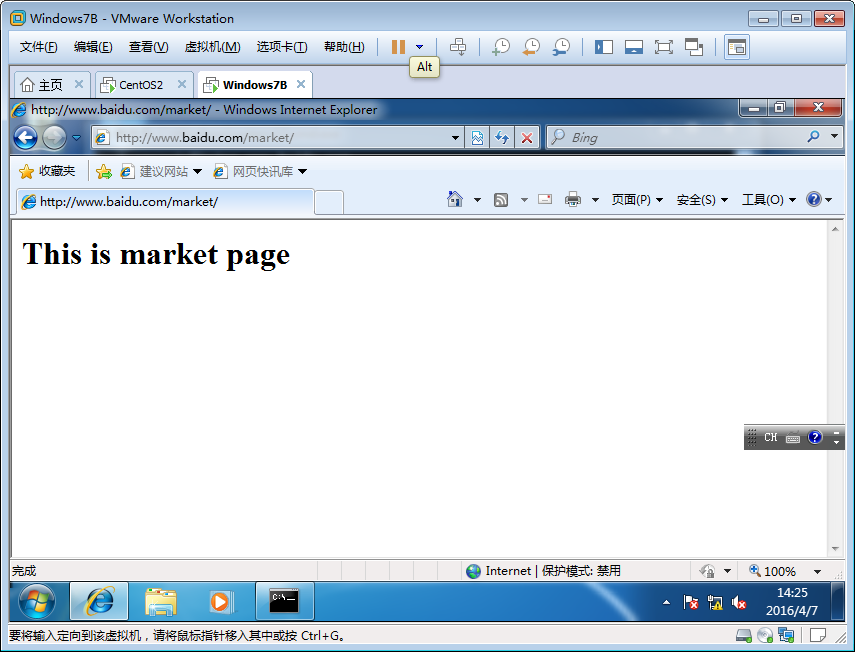
重启httpd服务



通过<http://www.baidu.com/s>访问



通过<http://www.baidu.com/market>访问虚拟目录



## 网站访问的控制

### 基于IP地址的控制

设置market目录只允许192.168.80.100和192.168.80.101访问

<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host.example.com

DocumentRoot /var/www/baidu

ServerName www.baidu.com

DirectoryIndex baidu.htm

ErrorLog logs/dummy-host.example.com-error\_log

CustomLog logs/dummy-host.example.com-access\_log common

<Directory "/var/www/baidu/market">

DirectoryIndex market.html

Order allow,deny

AllowOverride None

Allow from 192.168.80.100 192.168.80.101

#Allow from All 192.168.80.0/24 192.168.80.0/255.255.255.0允许所有

#172.16.0.0/16

</Directory>

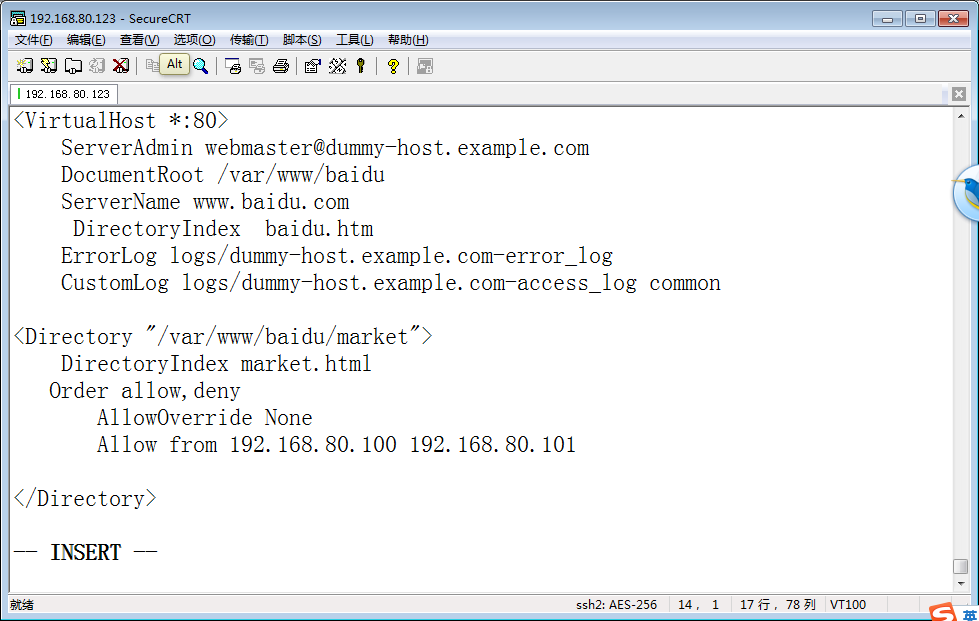
Alias /s /sales

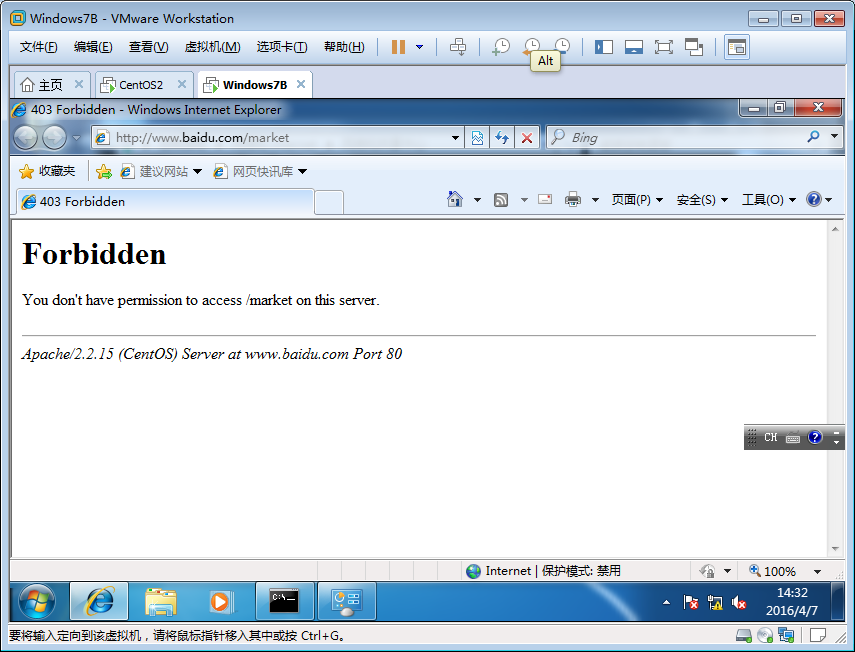
<DirectoryMatch /sales>

DirectoryIndex sales.html

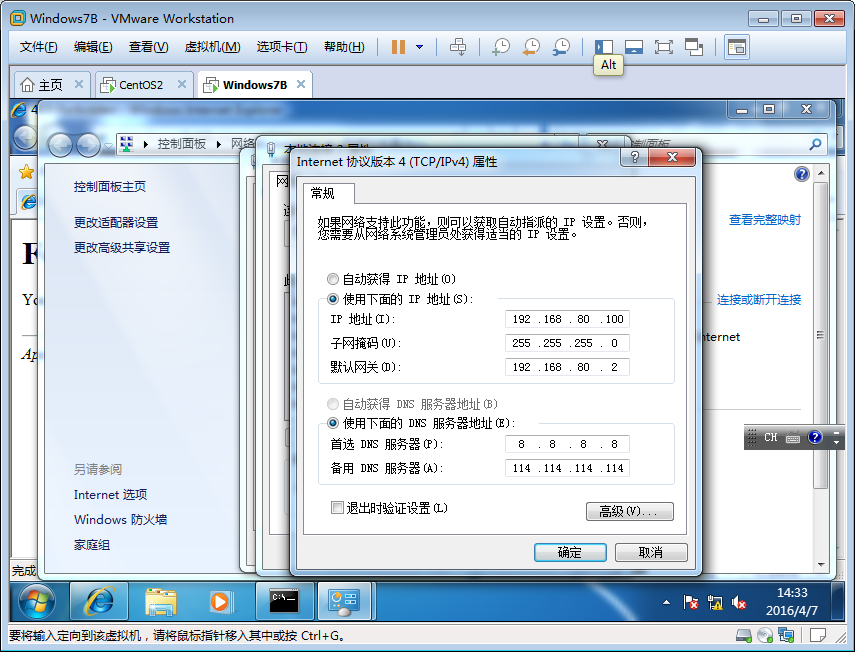
</DirectoryMatch>

</VirtualHost>

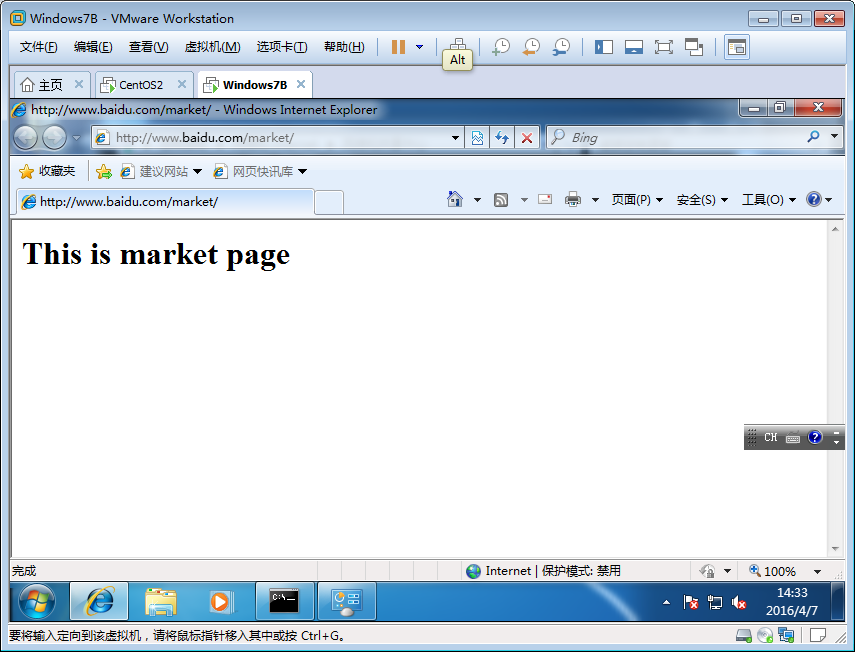




更改Windows7的IP地址



再次访问成功



## 实现身份验证

### 基本身份验证

创建身份验证文件 htpasswd建立和更新用于基本认证的存储用户名/密码的文本文件。

[root@CentOS2 ~]#mkdir /var/www/passwd

[root@CentOS2 ~]# cd /var/www/passwd/

[root@CentOS2 passwd]# htpasswd -c myrealm han

New password:

Re-type new password:

Adding password for user han

[root@CentOS2 passwd]# htpasswd myrealm wang

New password:

Re-type new password:

Adding password for user wang

[root@CentOS2 passwd]# ll

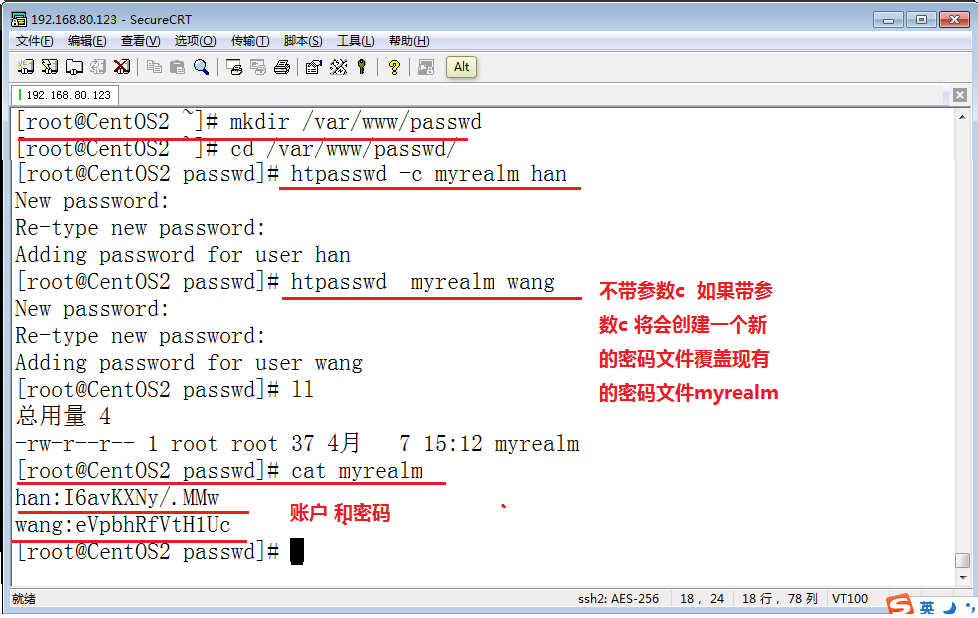
总用量 4

-rw-r--r-- 1 root root 37 4月 7 15:12 myrealm

[root@CentOS2 passwd]# cat myrealm

han:I6avKXNy/.MMw

wang:eVpbhRfVtH1Uc



将以下内容添加到[www.sogou.com](http://www.sogou.com)虚拟主机中，定义该目录身份验证方式和身份验证文件

<DirectoryMatch /var/www/sogou>

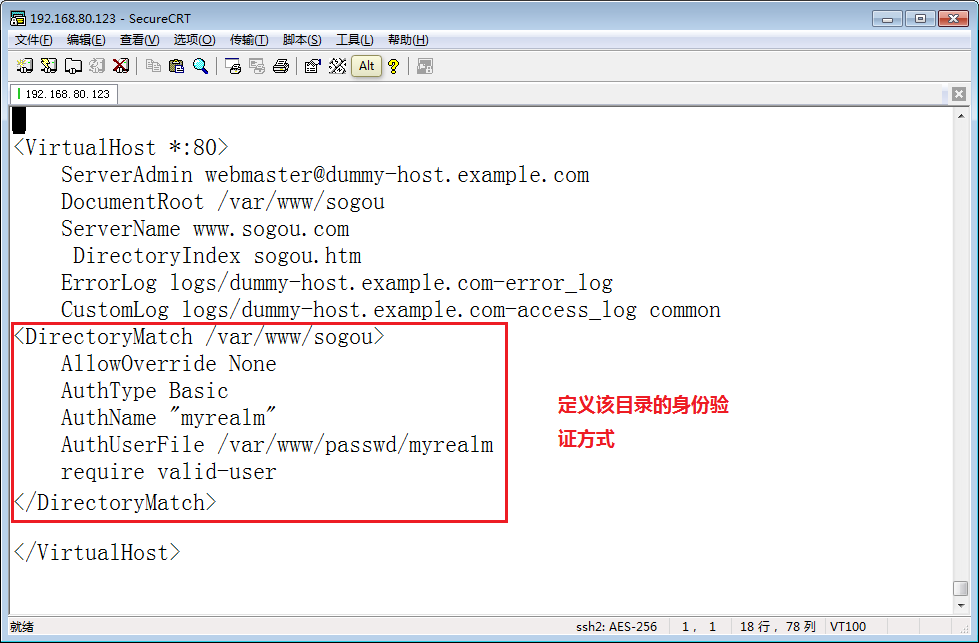
AllowOverride None

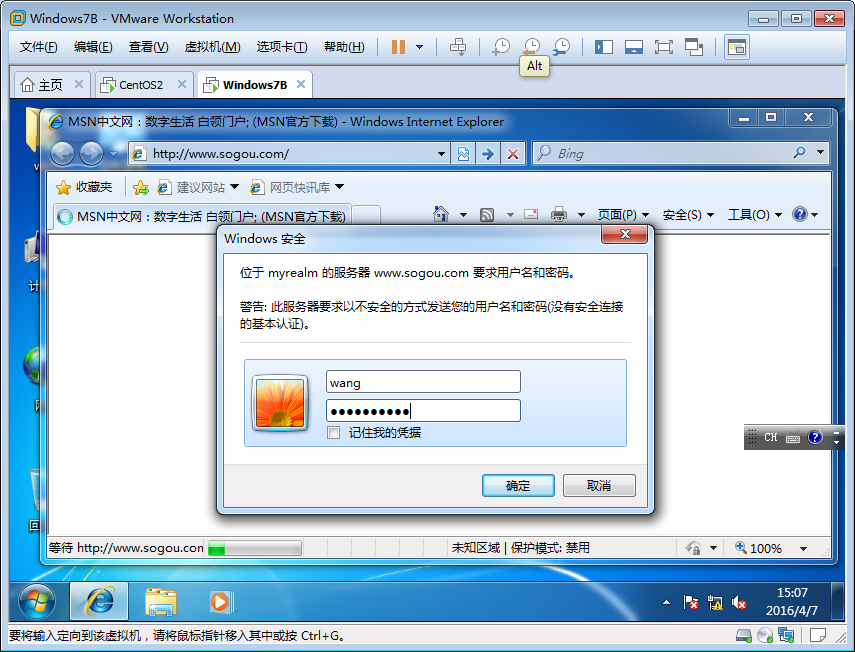
AuthType Basic

AuthName "myrealm"

AuthUserFile /var/www/passwd/myrealm

require valid-user

</DirectoryMatch>



### Digest身份验证

注意：红色部分和蓝色部分一定一直

[root@CentOS2 ~]# cd /var/www/passwd/

[root@CentOS2 passwd]# htdigest -c digestuser realm hanlg

Adding password for hanlg in realm realm.

New password:

Re-type new password:

Re-type new password:第二个用户已经去掉-c参数 否则覆盖前面创建的用户

[root@CentOS2 passwd]# htdigest digestuser realm wangrs

Adding user wangrs in realm realm

New password:

设置该文件的属主和权限

[root@CentOS2 passwd]# chown apache digestuser

[root@CentOS2 passwd]# chmod 700 digestuser

更改

<Directory "/var/www/html/sales">

AllowOverride None

AuthType Digest

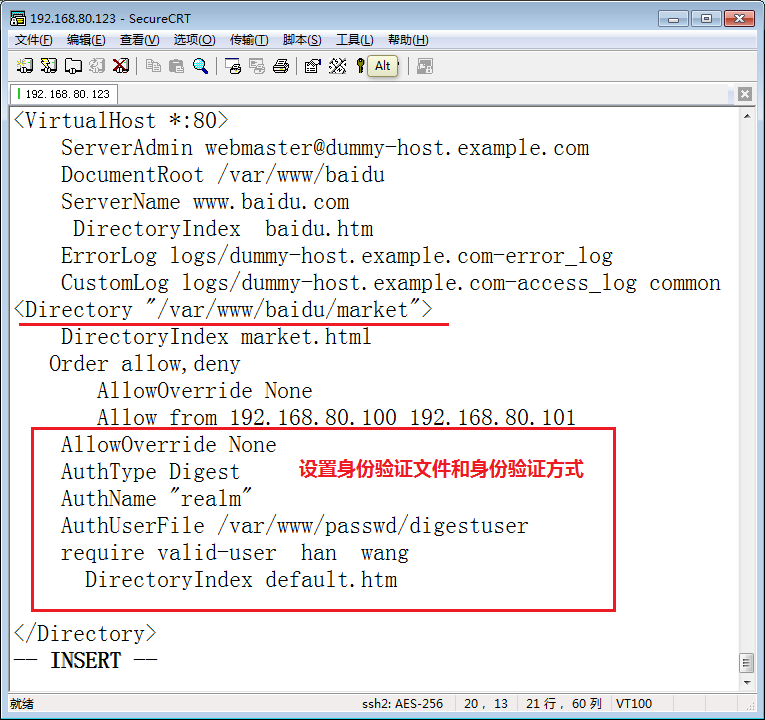
AuthName "realm"

AuthUserFile /var/www/passwd/digestuser

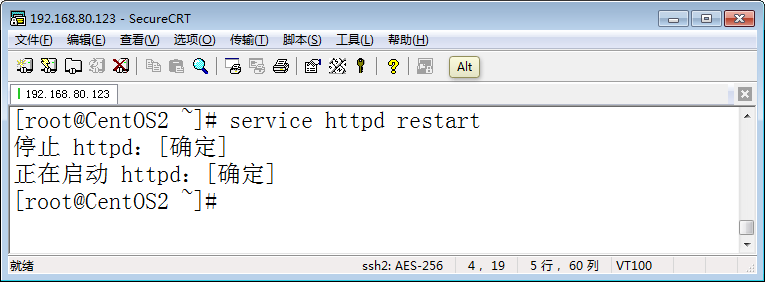
require valid-user han wang

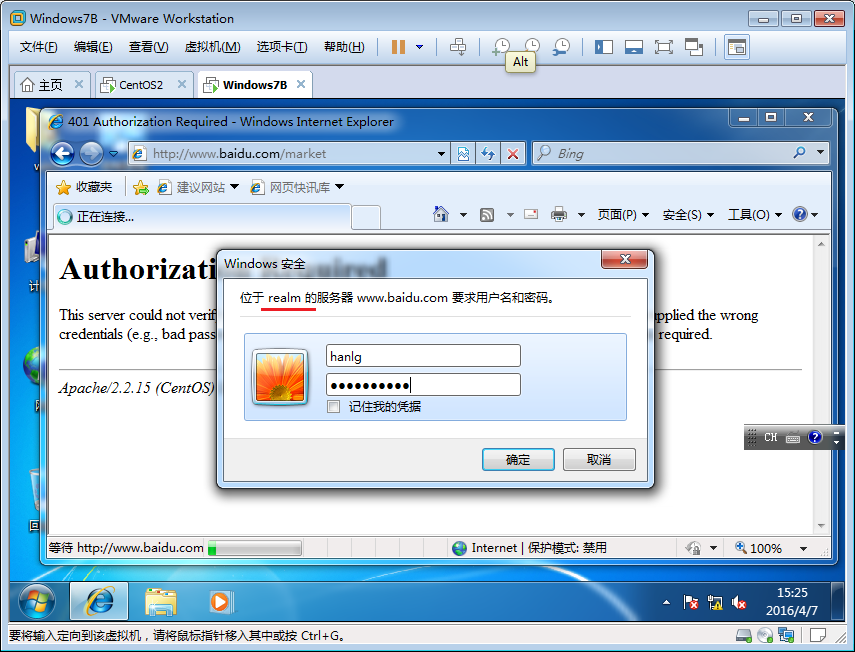
DirectoryIndex default.htm

</Directory>



重启httpd服务





## Apache日志

错误日志

访问日志

<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host.example.com

DocumentRoot /var/www/baidu

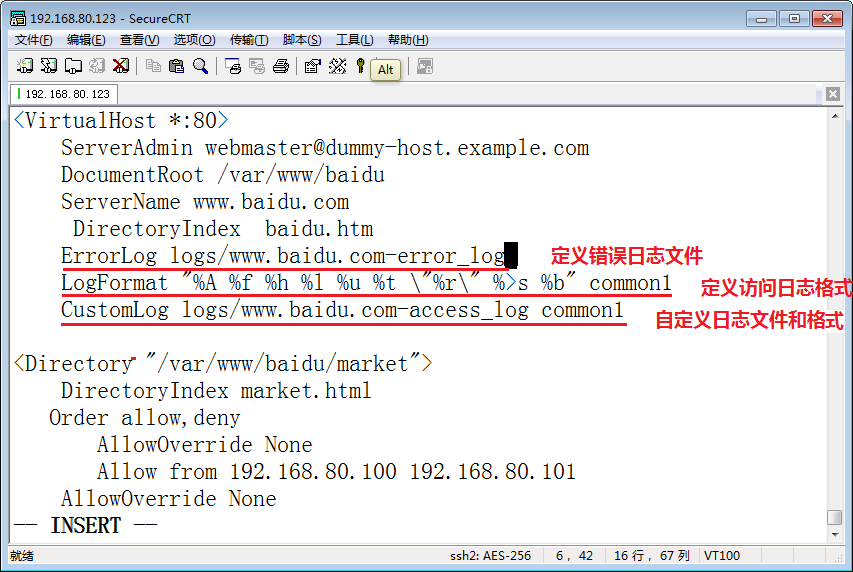
ServerName www.baidu.com

DirectoryIndex baidu.htm

ErrorLog logs/www.baidu.com-error\_log

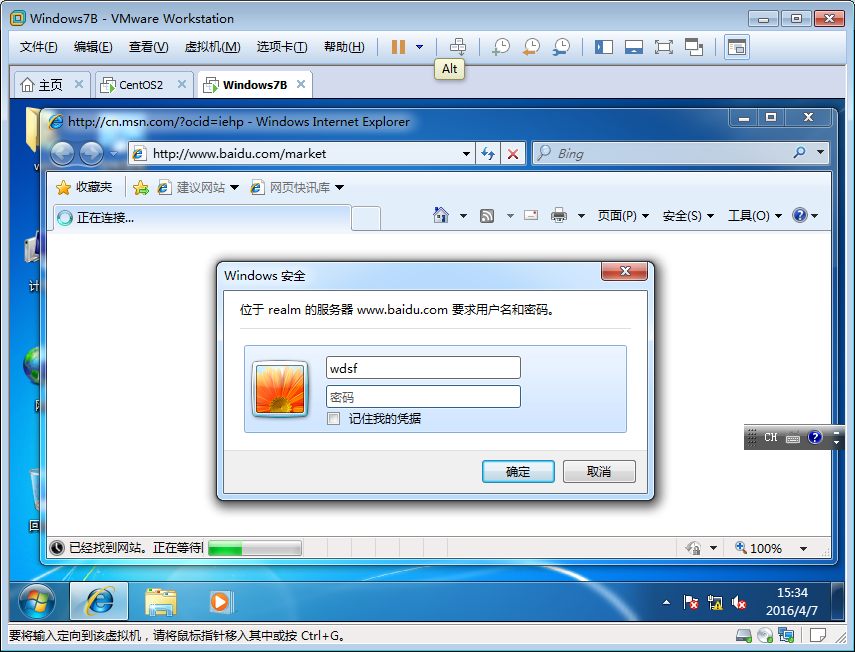
LogFormat "%A %f %h %l %u %t \"%r\" %>s %b" common1

CustomLog logs/www.baidu.com-access\_log common1



在Windows7上输入错误的账户登录www.baidu.com/market

再输入正确账号密码登录，这样能够产生错误日志和访问日志



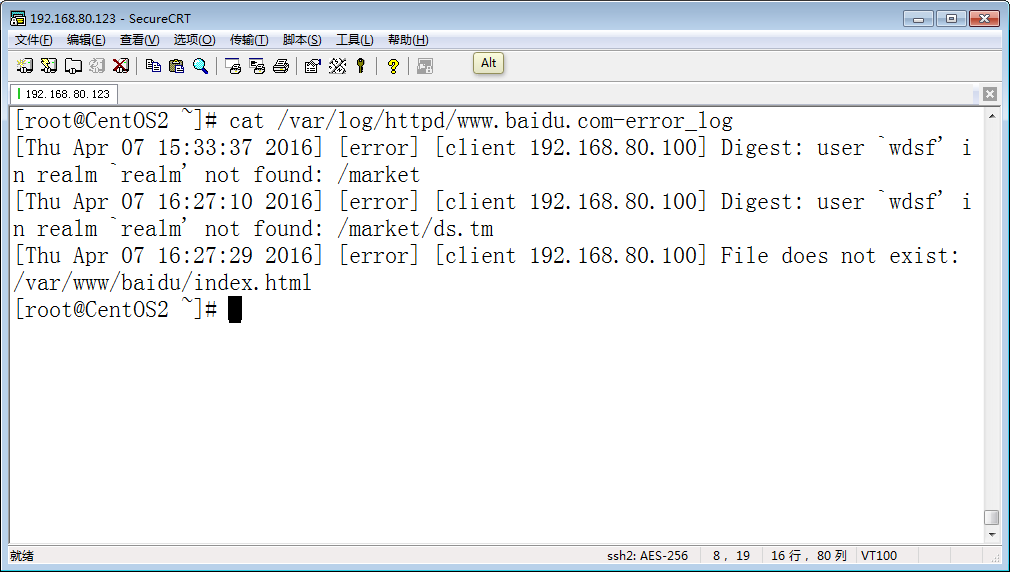
访问一个不存在的页面http://www.baidu.com/index.html



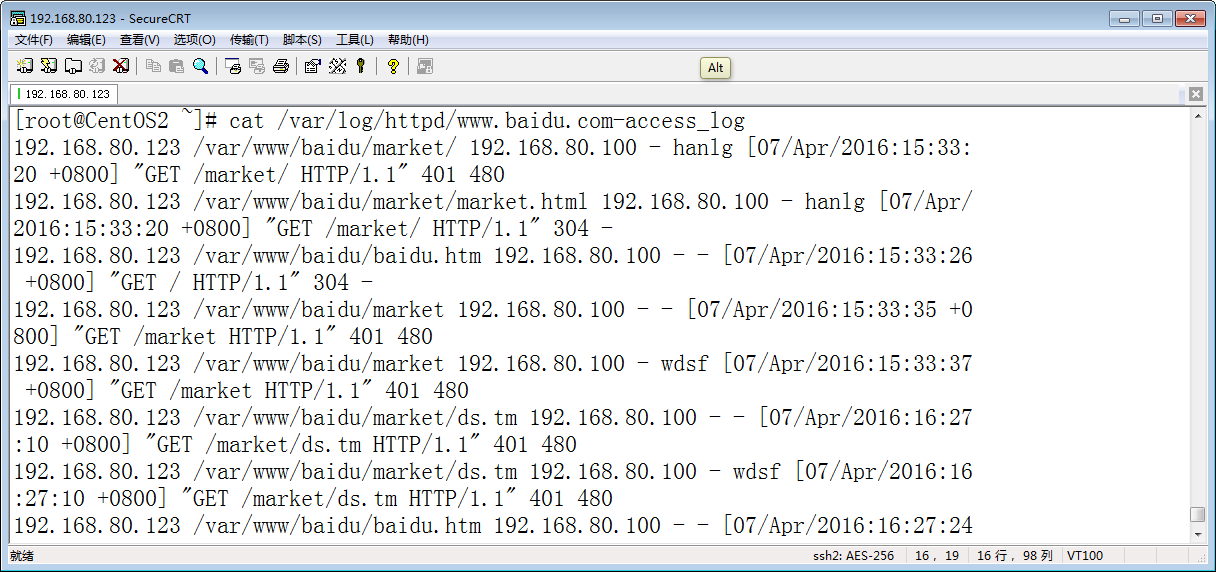
日志文件所在位置

/var/log/httpd

查看错误



查看访问日志



**apache日志记录格式的设置**  
定制日志文件的格式涉及到两个指令，即LogFormat指令和CustomLog指令，默认httpd.conf文件提供了关于这两个指令的几个示例。  
LogFormat指令定义格式并为格式指定一个名字，以后我们就可以直接引用这个名字。CustomLog指令设置日志文件，并指明日志文件所用的格式（通常通过格式的名字）。  
LogFormat指令的功能是定义日志格式并为它指定一个名字。例如，在默认的httpd.conf文件中，我们可以找到下面这行代码：  
LogFormat "%h %l %u %t \"%r\" %>s %b" common  
该指令创建了一种名为“common”的日志格式，日志的格式在双引号包围的内容中指定。格式字符串中的每一个变量代表着一项特定的信息，这些信息按照格式串规定的次序写入到日志文件。  
Apache文档已经给出了所有可用于格式串的变量及其含义，下面是其译文：  
%…a: 远程IP地址  
%…A: 本地IP地址  
%…B: 已发送的字节数，不包含HTTP头  
%…b: CLF格式的已发送字节数量，不包含HTTP头。例如当没有发送数据时，写入‘-’而不是0。  
%…{FOOBAR}e: 环境变量FOOBAR的内容  
%…f: 文件名字  
%…h: 远程主机  
%…H 请求的协议  
%…{Foobar}i: Foobar的内容，发送给服务器的请求的标头行。  
%…l: 远程登录名字（来自identd，如提供的话）  
%…m 请求的方法  
%…{Foobar}n: 来自另外一个模块的注解“Foobar”的内容  
%…{Foobar}o: Foobar的内容，应答的标头行  
%…p: 服务器响应请求时使用的端口  
%…P: 响应请求的子进程ID。  
%…q 查询字符串（如果存在查询字符串，则包含“?”后面的部分；否则，它是一个空字符串。）  
%…r: 请求的第一行  
%…s: 状态。对于进行内部重定向的请求，这是指\*原来\*请求 的状态。如果用%…>s，则是指后来的请求。  
%…t: 以公共日志时间格式表示的时间（或称为标准英文格式）  
%…{format}t: 以指定格式format表示的时间  
%…T: 为响应请求而耗费的时间，以秒计  
%…u: 远程用户（来自auth；如果返回状态（%s）是401则可能是伪造的）  
%…U: 用户所请求的URL路径  
%…v: 响应请求的服务器的ServerName  
%…V: 依照UseCanonicalName设置得到的服务器名字

### 查看Apache日志