一 配置环境

OS：CentOS 5.2

IP：172.20.9.24

计算机名：wang

域名：nainaitou.com

FQDN名：[www.nainaitou.com](http://www.nainaitou.com) ip：192.168.0.12

# 目前发现速度最快的CentOS5电信更新源！

|  |
| --- |
| 最近发现了一个国内电信用户最快的更新源。以前用163的更新源，但是163的CentOS更新不是很及时，而这个更新源更新比较及时，号称5个小时更新一次。 使用root帐号登录CentOS5，进行如下操作： cd /etc/yum.repos.d mv CentOS-Base.repo CentOS-Base.repo.save wget <http://centos.ustc.edu.cn/CentOS-Base.repo.5> mv CentOS-Base.repo.5 CentOS-Base.repo |

注：1.先说下用远程连接上去中文显示乱码的解决方法我看到几个人碰到了然后就选英文安装了：

我用的是putty来进行远程连接，只要把显示字体选成utf—8就可以了，putty上选到Windows-》Translation-》然后就看到个received data …不打全了 下面那个地方好选的选到utf-8就不会有乱码了 。

我们采用rpm方式安装

二 安装

首先看看要装点什么包

[root@localhost ~]# rpm -qa bind\*  
bind-sdb-9.3.4-10.P1.el5

bind-9.3.4-10.P1.el5

bind-libbind-devel-9.3.4-10.P1.el5

bind-libs-9.3.4-10.P1.el5

bind-utils-9.3.4-10.P1.el5

bind-devel-9.3.4-10.P1.el5

bind-chroot-9.3.4-10.P1.el5 [\\DN](file:///\\DN)S安全设定

[root@localhost ~]# rpm -qa caching-nameserver\*  
caching-nameserver-9.3.4-10.P1.el5 \\这个是DNS缓存

说明：

1.dns-chroot：没装dns-chroot的时候我们dns的配置文件的路径是/var/named就好了，装好了的话路径就变成/var/named/chroot/var/named 这样为了安全，要root权限才可以来改里面的配置

2.Caching-nameserver：这个事DNS的缓存包，装了你客户端查个你dns服务器上面没有的dns信息他会缓存起来，加快dns的效率。

安装的话命令用rpm –ivh bind\* 和rpm –ivh caching-nameserver

我是centos 我用yum install bind\* 和yum install caching-nameserver

注：2.在说个命令yum update 这个命令是把你里面的包都更新好，yum这个东西相当于ubuntu里面的apt都是软件源.

Ok装好了 让我们看下他们文件夹下面有点什么

三 Named.Conf文件配置

[root@localhost ~]# cd /var/named/chroot/etc/

[root@localhost etc]# ls

localtime named.caching-nameserver.conf named.rfc1912.zones rndc.key

啊 没有name.conf文件

好吧 我们新建个这个文件

[root@localhost etc]# vi named.conf

options {

directory "/var/named/"; //域名文件存放的绝对路径

forwarders {

202.96.199.133;

202.96.209.5;

168.95.1.1;

};

listen-on port 53 { any; };

};

zone "." {

type hint; //根域名服务器

file "named.ca"; //存放在//var/named/chroot/var/named目录，文件名为named.ca(有些地方也把这个文件命名为named.root他存放的是根服务器的地址

};

//下面定义localhost的正反解了

zone "localhost" {

type master; //类型为主域名服务器

file "named.localhost"; //本地正向解析的文件

};

zone "0.0.127.IN-ADDR.ARPA" {

type master;

file "named.127.0.0"; //本地反向解析的文件

};

//下面定义我自己的正反解了

zone "nainaitou.com" IN {

type master;

file "db.nainaitou.com"; //db.nai..这个名字你也可以用其他名字命名，我这样用db是为了和我们asiacorp保持一致

;};

zone "0.168.192.IN-ADDR.ARPA" {

type master;

file "named.192.168.0";

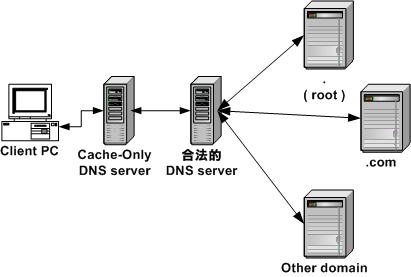
};

Ok写好了。

注：

3.我们看到我的设定里面有个forwards然后下面是我们熟悉的电信和hinet的dns，看到这2个地址你应该可以大致猜出来了吧，如果你的客户机要来查个dns 比如baidu.com而你的服务器上没有这条记录，那他就可以去外面的dns上面查询这条记录，你可以用这个来建立一个cache-only DNS server

工作原理如下：



配置的话那么你下面那些正反区域都不要了，可以就要到forwards下面的那些地址然后就可以了。

named.conf还没有完成

[root@localhost etc]# ls -l

总计 40

-rw-r--r-- 1 root root 405 05-27 01:53 localtime

-rw-r----- 1 root named 1195 01-21 20:27 named.caching-nameserver.conf

-rw-r--r-- 1 root root 423 05-27 03:26 named.conf

-rw-r----- 1 root named 955 01-21 20:27 named.rfc1912.zones

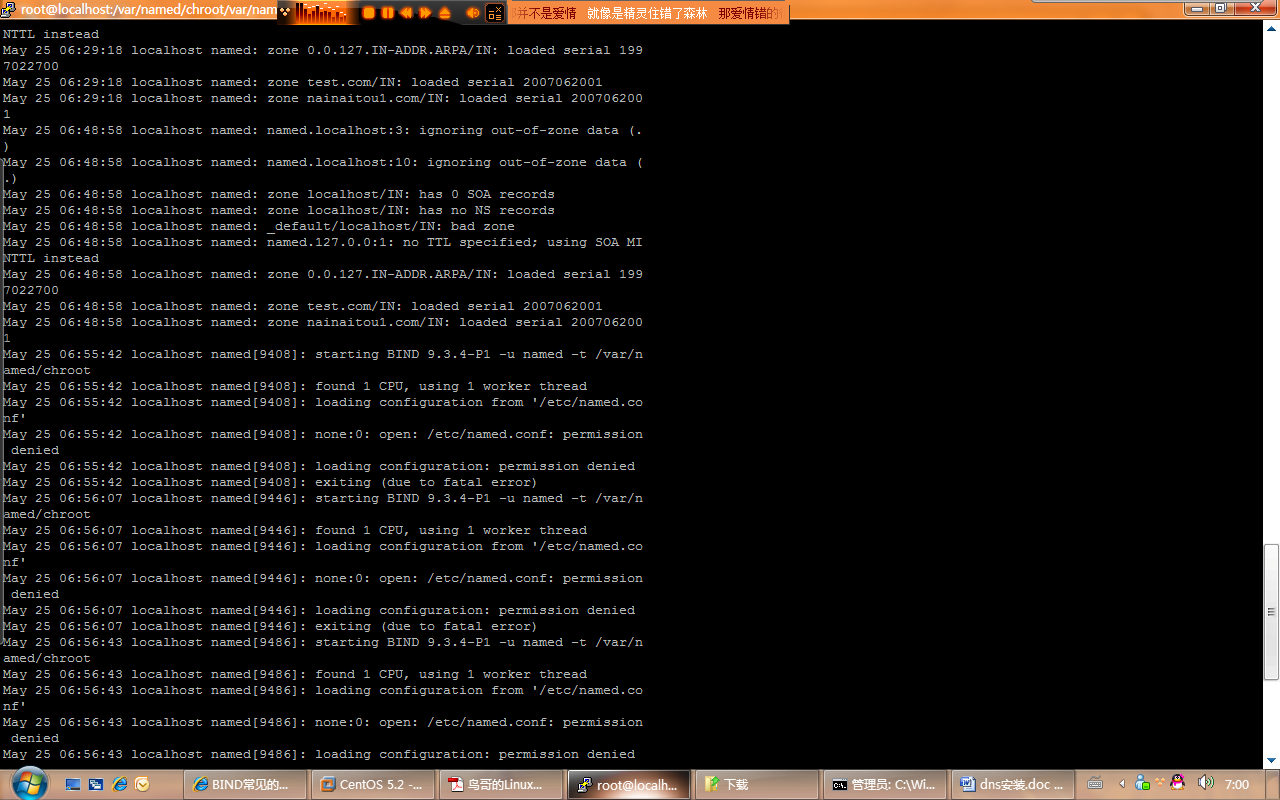
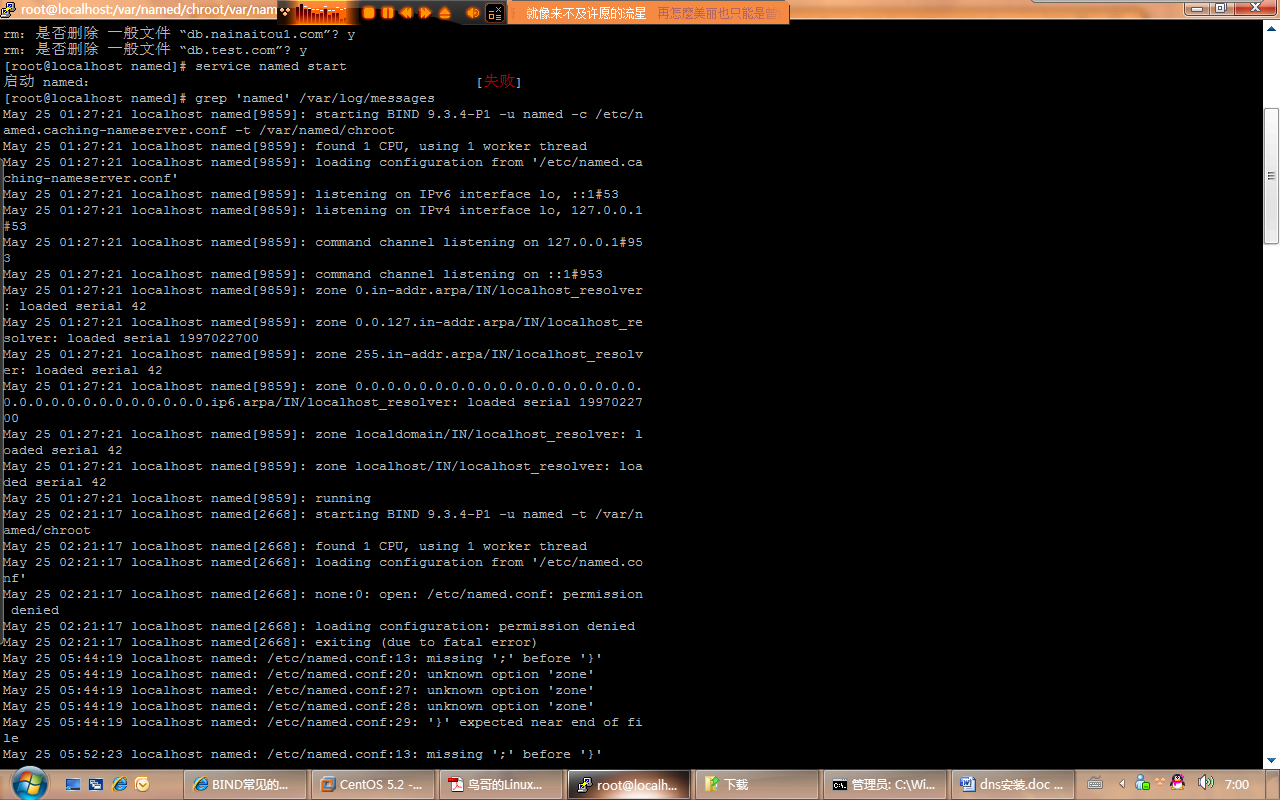
-rw-r----- 1 root named 113 05-27 03:15 rndc.key

看到了没有 group 别人都是named，而我们这个是root

这个一定要改过来 chgrp named named.conf

我原来设置的时候没有改过来，后来dns服务启动不起来，看下图：

没起来 然后我查看日志 grep ‘named’ /var/log/messages,看第二幅图



注意Open ：/etc/named.conf:permission denied

这个就是因为我没有改owned所以才会这样。

四 设定本机的正解档

[root@localhost named]# vi named.localhost

$TTL 86400 //这个跟清除 cache 的时间有关系！单位是秒！

@ IN SOA localhost. root.localhost. (

42 ; serial //Serial  与 master 及 slave 是否同步有关！

一般而言，如果这个数值变大了， slave 才会同步更新！

3H ; refresh //Refresh 定义出 slave 多久会主动的检查 serial 的值，以便主动的更新数据库！

15M ; retry //Retry   定义出，如果 slave 没有连上 master DNS 主机则多久之后会重新再次的主动检查！

1W ; expiry // Expire  如果一直没有连接上 mater ，那么到了这个时候 slave 就会放弃检查的动作了，不再更新！

1D ) ; minimum //Minimum 这个其实就是 TTL 啦！如果您没有定义 TTL ，   
 那么 TTL 的值就以这个来设定！

@ IN NS localhost. //特别留意最后面有个 . 喔！

localhost. IN A 127.0.0.1 //A 是正解里面 hostname 对应 IP 的标志

注：

5.符号说明

|  |  |
| --- | --- |
| $TTL | 1.这个东西定义出向外查询的数据可以记录在dns的cache中多久  2.后面的数字单位为秒  3. 通常这个数字如果定义太大的话，例如一天(86400)时，那么当别人更改了他的 DNS 讯息时，由于您的 cache 更新时间为一天，所以得要一天之后 cache 当中的数据才会被取代，因此，在一天之内，您查询到的信息『都会是旧的！』  4. 但是这个数字如果定的太小的话，例如五分钟(300)那么这部 DNS 将会不断的向外要求数据，则负荷会变的较大啦！  5其实，除非是在测试阶段，不然的话，通常都会建议定义一天的 cache 时间啰！ |
| @ | 这个就是 zone 定义出的那个咚咚啦！以这个档案内容为例，因为我们在 /etc/named.conf 当中就是定义出 localhost 这个 domain name 为一个 zone 的，因此，呵呵！在这里，这个符号就代表 localhost 啦！ |
| SOA | 1. 这个是 Start of Authority 开始设定的内容的意思啦！也就是接在后面的设定要开始了！请注意，这个咚咚在每个『zone 的设定档』当中都会存在！所以，每个 zone 的设定都一样即可！ 2. 在 SOA 后面会接两个咚咚，第一个为主机名称( localhost. )，请特别留意那个 localhost 后面有个小数点 (.) 这个东西很重要！他代表『一个完整的 hostname +  domain name 了』！如果没有加上 (.) 的话，那么就表示该文字『仅为 hostname ，还需要加上 domain name 』！这里是新手最容易出现的错误喔！第二个为管理员的 e-mail ！因为不能使用 @ (已经是特殊符号了)，所以这里也同样的以 (.) 来取代！例如上面我以 root@localhost 来做为我的 e-mail ，所以就写成了 root.localhost. ，同样的，最后面有个 (.) 喔！ 3. 在最后，会有小刮号 ( ) 括起了五个数字，这五个数字除了最后一个与 TTL 有关之外，其它的都跟 slave 与 master 的资料同步运作有关！    * + Serial ：这个数字仅是用来做为 master 与 slave 之间的 update 的参考数值也就是说，当 Slave 的 serial 小于 Master 时，那么 update 才会动作！由于担心设定者的设定技巧问题，因此通常我们以时间来做为 Serial 的订定依据，例如 2002 年 12 月 6 日第一次设定，可以写成『2002120601』请注意，这个数字不可超过 10 个数字。    * Refresh ：命令 slave 多久进行主动更新的时间；    * Retry ：如果到了 Refresh 的时间，但是 slave 却无法连接到 master 时，那么在多久之后，slave 会再次的主动尝试与主机联机；    * Expire ：如果 slave 一直无法与 master 连接上，那么经过多久的时间之后，则命令 slave 不要再连接 master 了！    * Minimun ：这个就有点像是 TTL 啦！ |
| Ns | 表示 name server 的意思 |
| A | 这是正解的符号啦！也就是说，前面的 localhost. (还是得要注意那个 . )所对应的 IP 为 127.0.0.1 的意思啦！ |
| . | 呵呵！再次的给他强调下，在 BIND 的设定档当中，关于主机名称的话，最后面有没有加上 . 是差很多的！加上了 . 表示这个『完整的主机名称，亦即是 hostname + domain name 』了，如果没有加上 . 的话，表示该名称仅为『 hostname 』而已！切记切记！ |

五 设定本地反解档

[root@localhost named]# vi named.127.0.0//这个文件名称是在 /etc/named.conf 里面设定

$TTL 600

@ IN SOA localhost. root.localhost. (

1997022700 ; Serial

28800 ; Refresh

14400 ; Retry

3600000 ; Expire

86400 ) ; Minimum

@ IN NS localhost.

1 IN PTR localhost.

上面前几行跟前面一样，就不提了！只有最后一行不一样，那个是什么呢？

注：

6.符号说明

|  |  |
| --- | --- |
| PTR | * 这是反解的符号啦！主要还是在于 IP 对应主机名称的咚咚！要注意的是，由于这个档案的 zone 为 127.0.0 ，所以我们只要加一个数字(最后一个数字)就可以啦！而那个 1 表示的就成为了 127.0.0.1 啰！ * 那么万一今天我们规划的是 B Class 的 zone 呢？例如 127.0 这样的 zone 呢？很简单啦！就填两个数字即可！也就是 0.1 啰！ * 最重要的东西就是：在规划 zone 的时候，是很重要的，而反解的 zone 的名称最后需要接上 in-addr.arpa，这点也请千万不要忘记了！ |

六 设定我自己的donmain name正解

[root@localhost named]# vi db.nainaitou.com

$TTL 3600

@ IN SOA ns.nainaitou.com. root.nainaitou.com. (

2007062001 ; Serial

10800 ; Refresh every 3 hours

3600 ; Retry every hour

604800 ; Expire after a week

3600 ) ; Minimum ttl of 1 hour

@ IN NS ns.nainaitou.com.

@ IN MX 10 mail.nainaitou.com.

www IN A 192.168.0.12

www IN TXT "this is a web server"

abc IN CNAME www

winxp IN A 192.168.0.100

winxp IN HINFO "INTEL-8300""Windows XP"

|  |  |
| --- | --- |
| SOA | * 特别注意到 SOA 那一行的设定喔！因为我们要设定的已经是 vbird.org 这个 Zone 了，所以请修改一下您的主机名称，还有 DNS 主机的管理员邮件地址喔！ * 另外，还是再次的提醒那个 . 是什么东西！ |
| MX | * 这个东西就是 Mail eXchanger (MX) 的简写，他的用途在使用于邮件主机时，需要的信件转递站！用于一般主机是没有多大的影响，但是对于 mail server 则有相当重要的影响哩！ * 请注意 MX 后面要接上一个数值喔！您可以设定多个邮件主机. |
| TXT | * 这个东西在进行『说明』而已！亦即是前面那部主机的一些信息。 * 特别注意的是，没事的话，『信息不要写得太详细，有的时候甚至应该要写些错误的讯息！』为什么呢？如果写得太详细的话，那么那些个 cracker 不就很简单的就可以将您的网站信息取得，并进而入侵了吗？ |
| CNAME | 这个东西就是设定主机别名的咚咚啦！因为我们的主机有很多个名字，没有必要为每个名字都建立一个 A 的标号，这个时候，我们就可以使用 CNAME 来设定另外一个别名！ |
| HINFO | 这个东西后面接两个咚咚，第一个接的是硬件的等级，第二个接的则是操作系统，这两个咚咚最好不要用在公开的 DNS 主机上面，跟 TXT 一样的问题啦！如果要设定的话，最好使用双引号分隔开来喔！ |

七 设定我自己的donmain name反解

设定反解要简单的多了～只要找到对应的 hostname 即可：

@ IN SOA nainaitou.com. root.nainaitou.com. (

2002120601 ; Serial

28800 ; Refresh

14400 ; Retry

720000 ; Expire

86400 ) ; minimum

@ IN NS nainaitou.com.

12 IN PTR nainaitou.com.

; The following is about other hosts

100 IN PTR winxp.nainaitou.com.

很简单把就是ip的对应即可

八 启动 named 与 port 及讯息确认

[root@localhost named]# service named restart

停止 named： [确定]

启动 named： [确定]

查看端口 没有问题

[root@localhost named]# netstat -tuln | grep 53

tcp 0 0 172.20.9.24:53 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:53 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:953 0.0.0.0:\* LISTEN

tcp 0 0 ::1:953 :::\* LISTEN

udp 0 0 172.20.9.24:53 0.0.0.0:\*

udp 0 0 127.0.0.1:53 0.0.0.0:\*

udp 0 0 0.0.0.0:5353 0.0.0.0:\*

udp 0 0 :::5353 :::\*

查看讯息 grep ‘named’ /var/log/message 注意红色的部分出错了

May 27 05:34:32 localhost named[10218]: starting BIND 9.3.4-P1 -u named -t /var/named/chroot

May 27 05:34:32 localhost named[10218]: found 1 CPU, using 1 worker thread

May 27 05:34:33 localhost named[10218]: loading configuration from '/etc/named.conf'

May 27 05:34:33 localhost named[10218]: listening on IPv4 interface lo, 127.0.0.1#53

May 27 05:34:33 localhost named[10218]: listening on IPv4 interface eth0, 172.20.9.24#53

May 27 05:34:33 localhost named[10218]: command channel listening on 127.0.0.1#953

May 27 05:34:33 localhost named[10218]: command channel listening on ::1#953

May 27 05:34:33 localhost named[10218]: named.127.0.0:1: no TTL specified; using SOA MINTTL instead

May 27 05:34:33 localhost named[10218]: zone 0.0.127.IN-ADDR.ARPA/IN: loaded serial 1997022700

May 27 05:34:33 localhost named[10218]: zone 0.168.192.IN-ADDR.ARPA/IN: loaded serial 2002120601

May 27 05:34:33 localhost named[10218]: zone nainaitou.com/IN: loaded serial 2007062001

May 27 05:34:33 localhost named[10218]: zone nainaitou1.com/IN: loaded serial 2007062001

May 27 05:34:33 localhost named[10218]: zone localhost/IN: loaded serial 42

May 27 05:34:33 localhost named[10218]: running

报错的内容no ttl 赶紧去找named.127.0.0.1这个文件发现时忘了 补上ok。

九 可能的设定错误

基本上，DNS 算是很难设定的一个 Server 了，所以在除错方面请务必小心！他的设定错误通常有两种情况：

* 语法设定错误：这个问题比较好解决，因为在 /var/log/messages 里面都已经说的很清楚了！按照内容去修订即可；
* 逻辑设定错误：这个就比较困扰了！为什么呢？因为他主要发生在您设定 DNS 主机的时候，考虑不周所产生的问题！例如忘记加上 (.) ，系统不会显示错误讯息，但是却会造成查询的误判，而 MX 设定的主机名称错误，也不会出现有问题的讯息，但是 mail server 就是会收不到信等等～这些错误都需要很详细的 DNS client 的测试才能知道问题的所在。

我们这里先就语法设定错误方面进行介绍，至于逻辑设定的问题，那个就需要多多的进行测试才能知道了～

我自己的那个错误的东西我记得我黏贴到文档上了，但是电脑没电了，没保存到，这里我就拿别人的报错来讲解了

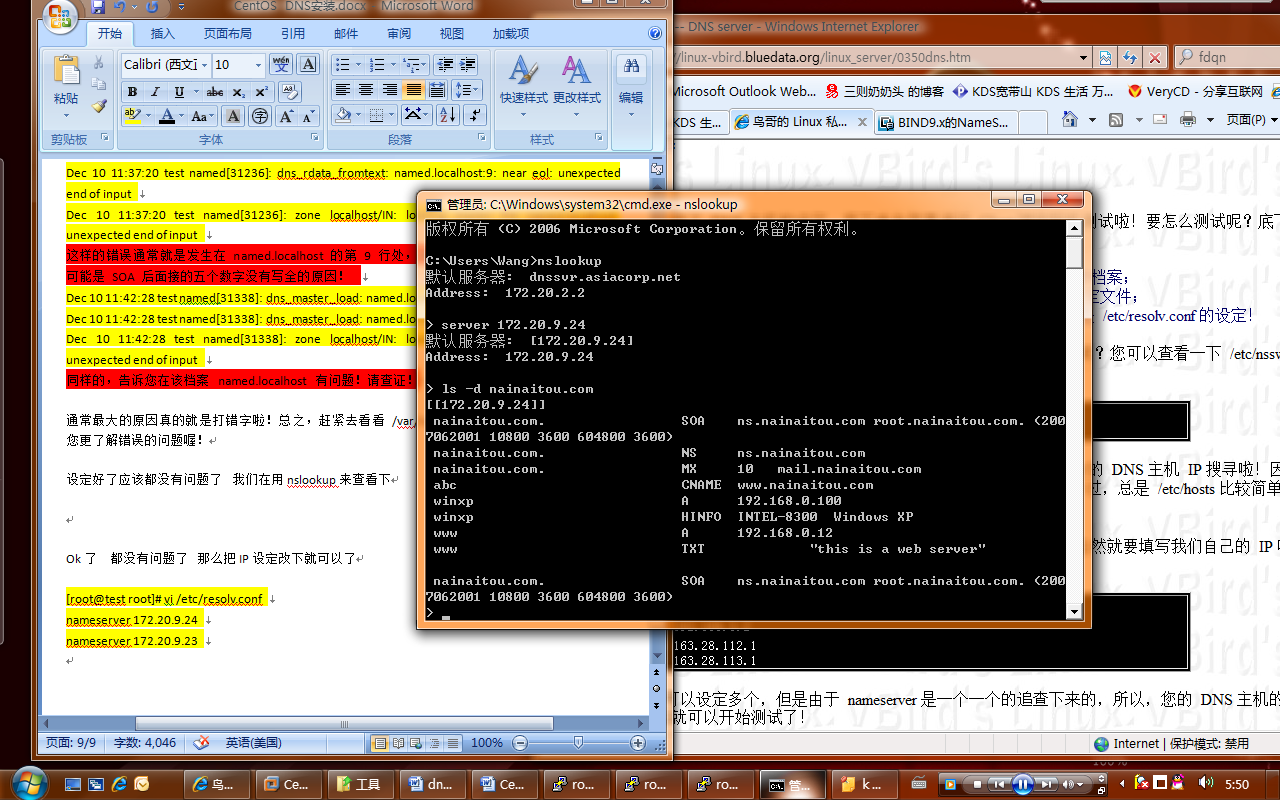
Dec 10 11:34:21 test named[31185]: loading configuration from '/etc/named.conf'   
Dec 10 11:34:21 test named[31185]: /etc/named.conf:18: missing ';' before '}'   
Dec 10 11:34:21 test named[31185]: loading configuration: failure   
Dec 10 11:34:21 test named[31185]: exiting (due to fatal error)   
这样的错误就是发生在 /etc/named.conf 的地 18 行，忘记加上 ; 符号了！去修正即可！

Dec 10 11:37:20 test named[31236]: dns\_rdata\_fromtext: named.localhost:9: near eol: unexpected end of input   
Dec 10 11:37:20 test named[31236]: zone localhost/IN: loading master file named.localhost: unexpected end of input   
这样的错误通常就是发生在 named.localhost 的第 9 行处，去修正看看，通常在前面几行很有可能是 SOA 后面接的五个数字没有写全的原因！   
Dec 10 11:42:28 test named[31338]: dns\_master\_load: named.localhost:14: unexpected end of line   
Dec 10 11:42:28 test named[31338]: dns\_master\_load: named.localhost:13: unexpected end of input   
Dec 10 11:42:28 test named[31338]: zone localhost/IN: loading master file named.localhost: unexpected end of input   
同样的，告诉您在该档案 named.localhost 有问题！请查证！

通常最大的原因真的就是打错字啦！总之，赶紧去看看 /var/log/messages 的内容，一定可以让您更了解错误的问题喔！

十 查询

设定好了应该都没有问题了 我们在用nslookup来查看下



Ok了 都没有问题了 那么把IP设定改下就可以了

[root@test root]# vi /etc/resolv.conf   
nameserver 172.20.9.24   
nameserver 172.20.9.23   
然后在把dns服务设置成开机启动

chkconfig --level 35 named on

完工

搭建备份dns服务器

直接把主DNS服务器的named.conf和/var/named/chroot/var/named配置拷了过来修改。

把type master改成type slave，添加主DNS服务器的地址  
masters { 172.20.9.24; };

options {

directory "/var/named/";

forwarders {

202.96.199.133;

202.96.209.5;

168.95.1.1;

};

listen-on port 53 { any; };

};

zone "." {

type hint;

file "named.ca";

};

zone "localhost" {

type slave;

file "named.localhost";

masters { 172.20.9.24; };

};

zone "0.0.127.IN-ADDR.ARPA" {

type slave;

file "named.127.0.0";

masters { 172.20.9.24; };

};

zone "nainaitou.com" IN {

type slave;

file "db.nainaitou.com";

masters { 172.20.9.24; };

;};

zone "0.168.192.IN-ADDR.ARPA" {

type slave;

file "named.192.168.0";

masters { 172.20.9.24; };

};

**可能的错误**，注意了 我做完后更新nainaitou.com 但是老是不成功

用grep ’ named’ /var/log/message 显示下面的错误

[named: dumping master file: tmp-xxxxxxx: open: permission denied](http://forums.fedoraforum.org/archive/index.php/t-106208.html)

然后网上找了下 原来你要把/named这个文件夹的group改成named否则没权限写入

Ok了 全部都完成了

快速添加dns脚本

我们要加dns可以直接用一条dnsadd的命令，但是这条命令原来是不带的，要我们建一个shell脚本。

一般shell脚本都在/usr/bin 目录下面

进入后 输入vi dnsadd

然后把下面的内容黏贴进去

下面这个是添加到主DNS里去的脚步，我再原来我们2.17的脚步上修改了下

添加前先要到/var/named/chroot/var/named下面新建个文件名字叫asmod（做为dns的模版）

Asmod内容如下你可以根据自己需要修改：

$TTL 3600

@ IN SOA ns.nainaitou.com. root.nainaitou.com. (

2007062001 ; Serial

10800 ; Refresh every 3 hours

3600 ; Retry every hour

604800 ; Expire after a week

3600 ) ; Minimum ttl of 1 hour

IN NS ns.nainaitou.com.

@ IN MX 10 mail.nainaitou.com.

www IN A 192.168.0.12

**脚步**

#/bin/sh

//上来先定义下 下面2个namedef和dbdir的文件路径

Namedef=/etc/named.conf

Dbdir=/var/named/chroot/var/named

if

[ $# != 1 ]

then

echo "usage: dnsadd [domain name]"

else

if

grep $1 $Namedef //找下这个位置新建的域名是不是已经存在了

then

echo This domain has been exist! //已经存在了就显示这句话

else

if

test "$(ls $Dbdir | grep db.$1)" //也去dbdir这个路径检查下

then

echo "Dbfile db.$1 has been exist, please check again!"

else

cp "$Dbdir"/asmod "$Dbdir"/db.$1; //检查了都没有那么开始新建了到我们那个asmod模版复制去了

chown named:named "$Dbdir"/db.$1 //复制好了还改变下权限

echo "zone \"$1\" IN {type master;file \"db.$1\";allow-update{none;};};" >>$Namedef // 还写到named.conf文件里

/usr/sbin/rndc reconfig

if

[ $? = 0 ]

then

echo Domain $1 record has been added successful! //显示添加成功

else

echo Please check your input again! Named restart failed!

fi

fi

fi

fi

下面这个是给slave的脚步

#/bin/sh

Namedef=/etc/named.conf

Dbdir=/var/named/chroot/var/named

if

[ $# != 1 ]

then

echo "usage: dnsadd [domain name]"

else

if

grep $1 $Namedef

then

echo This domain has been exist!

else

if

test "$(ls $Dbdir | grep db.$1)"

then

echo "Dbfile db.$1 has been exist, please check again!"

else

cp "$Dbdir"/asmod "$Dbdir"/db.$1;

chown named:named "$Dbdir"/db.$1

echo "zone \"$1\" IN {type slave;file \"db.$1\";masters {172.20.

2.17;};};" >>$Namedef

/usr/sbin/rndc reconfig

if

[ $? = 0 ]

then

echo Domain $1 record has been added successful!

else

echo Please check your input again! Named restart failed

!

fi

fi

fi

fi

**Dns配置高级篇**

DNS的配置文件named.conf是有非常多的可选项的，这里只是介绍一点常用的配置。如果你觉得还想继续研究下去，那也不需要在网上到处找资料的，直接用“man named.conf”就可以得到一个非常非常详细的说明。  
  
1．Options 我们通过options可以定制一个性能更优、安全性更高的dns服务器。  
  
Version “I am nainaitou”;           
  
别人想探测我们dns版本，然后根据该版本的漏洞来攻击我们。休想！配置了这条命令后，别人再探测的版本后就是“I am cpss”了，呵呵。  
  
Allow-transfer {192.168.1.1;192.168.1.4;};  
  
如果没有配置这一条命令，任何人都可以通过nslookup工具来得到你域里面的zone文件，也就是说他得到了你的主机列表，然后再分析，再……。当然，slave dns需要你允许它能够传送，否则它就得不到master dns上的zone文件，也就没办法工作了。这里假设192.168.1.1和192.168.1.4是该dns服务器的slave服务器，在master服务器上配置了如上命令。  
  
Listen-on{192.168.1.2;};  
  
增加上这条命令，启动dns时就不会监听所有网络接口的53端口了，只监听指定网络接口的53端口。  
  
Blackhole {hatenets;};  
  
我们不想让某些网段使用我们的dns服务器，就用这条命令吧。不过还需要配置一个acl来定义匹配的网段，如下所示：  
  
acl hatenets {  
  
1.0.0.0/8;  
  
2.0.0.0/8;  
  
};  
这两个网段的地址是无法使用我们的dns了。  
  
  
  
  
  
2．logging 通过该选项，我们可以生成我们想要的日志。通过日志，我们可以更好地维护dns服务器。  
  
Logging {  
  
        Channel syslog\_info {  
  
        File  “/var/log/bindall.log” versions 20 size 2m;  
  
        Print-category yes;  
  
        Print-time yes;  
  
        Severity notice;  
  
                      };  
  
        category         default {  
  
               syslog\_info;  
  
                                    };  
  
        };  
上例中我们建立了一个安全级别为notice的日志，dns的报错信息都会存放在“/var/log/bindall.log”日志文件了，分析该文件，我们就能轻松找出dns工作不正常的原因了。  
  
Logging里面的安全级别很多，大家可以根据自己需要来制定一个或多个logging。下面是logging的语法：  
  
     logging {  
  
            [ channel channel\_name {  
  
              ( file path\_name  
  
                 [ versions ( number | unlimited ) ]  
  
                 [ size size\_spec ]  
  
               | syslog ( kern | user | mail | daemon | auth | syslog | lpr |  
  
                          news | uucp | cron | authpriv | ftp |  
  
                          local0 | local1 | local2 | local3 |  
  
                          local4 | local5 | local6 | local7 )  
  
               | null );  
  
  
  
              [ severity ( critical | error | warning | notice |  
  
                           info  | debug [ level ] | dynamic ); ]  
  
              [ print-category yes\_or\_no; ]  
  
              [ print-severity yes\_or\_no; ]  
  
              [ print-time yes\_or\_no; ]  
  
            }; ]  
  
  
  
            [ category category\_name {  
  
              channel\_name; [ channel\_name; ... ]  
  
            }; ]  
  
            ...  
  
          };

|  |  |
| --- | --- |
| A | 一个主机地址 |
| AAAA | IPv6地址 |
| A6 | IPv6地址。它可以是一部分地址（一个后缀）加上一个可以找到其余部分地址（前缀）的间接指向的名字。试验性的。 |
| AFSDB | AFS数据库服务器的位置。试验性的。 |
| APL | 地址后缀列表。试验性。 |
| CERT | 存放一个数字签名。 |
| CNAME | 指定一个别名的正规名。 |
| DNAME | 使用另一个名字替代所指定的名字后进行查找，有效地将域名空间的一个完整子域作为别名，与之相对的使CNAMERR |
| DNSKEY | 储存与一个签名DNS区相关的公钥。 |
| DS | 储存与一个签名DNS区相关的公钥的hash。 |
| GPOS | 指定全球位置。为LOC所替代。 |
| HINFO | 指定主机所用的CPU和OS。 |
| IPSECKEY | 提供一个储存DNS中的Ipsec keying材料的方法。 |
| ISDN | ISDN地址的表示。试验性的。 |
| KEY | 存放一个与DNS名字相关的公钥。用于最初的DNSSEC；现被DNSSECbis中的DNSKEY所替代，但仍然用于SIG（0） |
| KX | 指定一个DNS名字的密钥交换者。 |
| LOC | 存放GPS信息。 |
| MX | 为一个域指定邮件交换者。由邮件交换者的主机名后跟一个16位的优先数（数值越小越优先）。 |
| NAPTR | 名字权威指针。 |
| NSAP | 一个网络服务访问点。 |
| NS | 域的权威名字服务器。 |
| NSEC | 用于DNSSECbis中，指明在某个名字内部并带有属主名的资源记录不再一个区内，也指明某个存在的名字所提供的资源记录类型。 |
| NXT | 用于DNSSEC中，安全地指明在某个名字之间的给定属主名的资源记录不存在于一个区中，并指明对一个存在的名字具有哪种类型的资源记录。在原始的DNSSEC中使用；在DNSSECbis中被NSEC取代。 |
| PTR | 指向域名空间另一部分的指针。 |
| PX | 在RFC 822和X.400地址之间提供映射。 |
| RP | 关于域的负责人信息。试验性的。 |
| RRSIG | 包含DNSSECbis签名数据。 |
| RT | 绑定在没有自己的直接的广域网地址的主机上的路由出口。 |
| SIG | 包含DNSSEC签名数据。在原始的DNSSEC中使用；已被DNSSECbis中的RRSIG所替代，但是仍然用作SIG（0）。 |
| SOA | 标识一个授权区的头部。 |
| SPF | 包含一个给定电子邮件域名的发生者策略框架信息。 |
| SRV | 关于众所周知的网络服务的信息(替代WKS)。 |
| SSHFP | 提供发布一个安全shell密钥指纹的安全方式。 |
| TXT | 文本记录。 |
| WKS | 关于一个域所支持的众所周知的网络服务的信息，如SMTP |
| X25 | X.25网络地址的表示。试验性的。 |