在Linux下轻松搭建自己的DNS服务器 - 爱维Linux运维培训 - 51CTO技术博客

经常上网的朋友可能经常去新浪、搜狐等大型网站，只需要在浏览器输入它们的网址即可实现访问，看似非常简单，但是从技术层面来讲，却包含了一个复杂的过程：在访问网页的时候，首先在浏览器输入网站域名，接着浏览器会根据本机DNS服务器的设置将输入的网站域名转换为对应的IP地址，然后才去这个IP对应的服务器上请求数据，最后将请求得到的数据通过浏览器显示出来。这个过程最主要的一个环节就是从域名到IP地址的转换，而这个工作就是靠DNS服务器实现的。

一、 DNS服务概述

DNS是Domain Name System的缩写,即域名系统,DNS服务主要的功能是将域名转换为相应的IP地址，提供DNS服务的系统就是DNS服务器。

DNS服务器可以分为3种，主域名服务器（Master DNS）、辅助域名服务器（Slave DNS）和高速缓存服务器（Cache-only server）。

Master DNS，本身提供dns服务，并且本身含有区域数据文件。

Slave DNS，和Master一起提供dns服务，当Master服务器上的配置信息修改的时候，会自动更新到Slave服务器达到同步。

Cache-only server，没有自己的区域数据文件，只是帮助客户端向外部dns请求查询，然后将查到的结果保存到它的缓存中。

在linux系统下DNS服务的功能是通过bind软件实现的，几乎每个linux发行版都自带了这个DNS服务软件，下面将具体讲述DNS服务的安装、配置与使用。

二、 DNS服务的搭建

这里我们的讲述环境为：

操作系统：Red Hat Enterprise Linux Server release 5

bind软件版本：系统自带bind-9.3.4

1. 安装bind软件

Rhel5系统下安装bind需要同时安装bind-utils、bind-chroot、ypbind、bind-libs、caching-nameserver几个支持bind的软件包。检查系统是否正确安装了bind软件，执行如下命令：

```
[root@localhost ~]# rpm -qa |grep bind
bind-libs-9.3.4-6.0.2.P1.el5_2
bind-utils-9.3.4-6.0.2.P1.el5_2
bind-chroot-9.3.4-6.0.2.P1.el5_2
ypbind-1.19-8.el5
bind-9.3.4-6.0.2.P1.el5_2
[root@localhost ~]# rpm -qa |grep caching-nameserver
caching-nameserver-9.3.4-6.0.2.P1.el5_2
```

上面的几个软件包都可以从系统安装光盘找到，如果没有安装或者缺少某些包，请自行通过rpm方式进行安装，这里不在过多讲述。

如果你的系统支持yum方式自动升级，只需执行如下命令即可自动完成安装：

```
[root@localhost ~]#yum install bind caching-nameserver
```

2. 配置DNS服务

Bind软件在rhel 5版本中使用了chroot技术，与其它linux版本下的配置不尽相同，例如没有DNS服务的核心配置文件named.conf以及任何区域数据文件，安装程序的路径也与其它版本不同。不过这些并不影响我们对DNS的配置，下面首先讲述bind在rhel5下的安装目录结构。

Bind安装完毕，主程序目录默认为/var/named，由于rhel5下的bind默认安装后没有named.conf文件，而我们在上面安装了caching-nameserver包，这个包提供了rhel5下bind的初始化模板文件，所以/var/named/chroot/etc是DNS的核心配置文件目录，/var/named/chroot/var/named目录则是系统自带的区域数据文件及自己建立的区域数据文件的位置。

（1）named.conf文件详解

这里我们通过系统提供给bind的初始化模板文件构造出named.conf文件来。

```
[root@localhost ~]#cd /var/named/chroot/etc
[root@localhost etc]# cp named.rfc1912.zones  named.conf
[root@localhost etc]#chown root:named named.conf
```

在这里，我们通过拷贝named.rfc1912.zones文件构造出了named.conf主配置文件。然后将named.conf的权限设置为root:named，注意，这个授权很重要，要不然DNS无法正常工作。

```
[root@localhost ~]#vi /var/named/chroot/etc/named.conf
// named.rfc1912.zones:
// Provided by Red Hat caching-nameserver package
// ISC BIND named zone configuration for zones recommended by
// RFC 1912 section 4.1 : localhost TLDs and address zones
// See /usr/share/doc/bind*/sample/ for example named configuration files.
```

在named.conf配置文件中主要使用“//”和“/* */”来进行注释。

```
options {
directory     "/var/named";
};
```

上面这段通过OPTIONS选项定义了一些影响整个DNS服务器的环境设置，directory选项指定named从/var/named目录下读取DNS数据文件，这个目录用户可自行指定并创建，指定后所有的DNS数据文件都存放在此目录下。

```
zone "ixdba.net" IN {
        type master;
        file "ixdba.net";
        allow-update { none; };
};
```

上面这段设置是用zone关键字来定义一个正向域区，对应的域名分别为ixdba.net，一个zone关键字定义一个域区。在这里type类型有三种，它们分别是master,slave和hint，它们的含义分别是：

Master：表示定义的是主域名服务器。

slave：表示定义的是辅助域名服务器。

hint：表示是互联网中根域名服务器。

file用来指定存放DNS记录的文件，allow-update定义是否允许客户主机或服务器自行更新DNS记录，上面指定的这个正向区域不允许更新DNS记录。

```
zone "60.168.192.in-addr.arpa" IN {
        type master;
        file "60.168.192.zone";
        allow-update { none; };
};
```

上面这段设置是定义一个IP为192.168.60.*的反向区域。

（2）区域数据文件的设定

在/var/named/chroot/var/named目录下，我们定义出上面指定的几个区域数据文件。

```
[root@localhost ~]#cd /var/named/chroot/var/named
[root@localhost named]#cp localhost.zone  ixdba.net
[root@localhost named]#cp named.local  60.168.192.zone
[root@localhost named]#chown root:named ixdba.net 60.168.192.zone
```

下面我们分析下正向区域数据文件的格式和含义，主要看下我们已经设定好的ixdba.net区域数据文件：

```
[root@localhost named]#more /var/named/chroot/var/named/ixdba.net
$TTL    86400
@               IN SOA  ns.ixdba.net.     root.ixdba.net.(
                                        42              ; serial (d. adams)
                                        3H              ; refresh
                                        15M             ; retry
```

```
                                        1W              ; expiry
                                        1D )            ; minimum
                IN NS           ns.ixdba.net.
        IN MX 10  mail
                IN A            192.168.60.133
ns              IN A            192.168.60.133
www     IN A       192.168.60.135
mail     IN A       192.168.60.136
linux    IN CNAME  www
```

可以看出，区域数据文件内容很简单。

第一行是一个TTL设定，定义区域数据文件里面的各项记录的默认TTL值为86400 秒，缺少此行不影响使用，但是会出现警告信息。

第二行，是一个SOA记录的设定，“@”代表相应的域名，也就是在named.conf中设定的zone，如在这里表示ixdba.net，IN表示后面的数据使用的是INTERNET标准。SOA，全称是“Start Of Authority”的意思，表示目前区域授权开始。每一个区域数据文件只能有一个SOA，不能重复，而且必须是所负责的zone中第一个“记录”。在SOA后面分别指定了这个区域的授权主机名称和管理者的信箱，特别注意，授权主机名和管理员信箱后面都要有一个“.”，而且授权主机名称必须能够在DNS设置中找到一个A记录（下面会讲到），由于“@”在区域数据文件中有其它含义，因此管理员信箱邮件地址中用“.”代替“@”符号。

接下来包含在括弧中的5组数字是作为与slave服务器同步信息而设置的，含义如下：

Serial：表示配置文件的修改版本，格式是年月日加上修改的次数，每次修改这个配置文件时都应该修改这个数字，因为slave DNS进行信息同步时，会比较这个数值，如果这个数值比自身的数值大，那么就进行更新，否则忽略更新。注意，这个设置很重要，如果你在修改区域数据文件后，没有更新该值，那么所作的更改就不会更新到网上的其它DNS服务器。

refresh：用来设定slave DNS与Master DNS进行同步的间隔时间。

retry：如果slave DNS在进行更新失败后，要隔多久再进行重试。

expiry：设定slave DNS在与Master DNS同步失败后，多长时间清除对应的记录。

Minimum：这是默认的最小TTL值，如果在前面没有指定TTL值，就以这个为基准。

以上的数字都是以秒为单位，但也可以用 H(小时)、D(天)、W(星期)来做单位。

第8至14行，是对域名解析的具体设置，第一列表示不同的主机域名，但是省略了后面的域信息。例如“www”其实是www.ixdba.net，“mail”是指mail.ixdba.net。其它具有相同的含义。“IN”后面的指令含义说明如下：

NS：用来定义这个主机是个域名服务器。

MX：定义了一个邮件交换器。

A指针：定义了一个A记录，即域名到IP的记录。

CNAME：定义了域名的别名。

从上面的例子可知，我们首先定义了一个NS（name server）为ns.ixdba.net，然后定义了一个邮件交换器，交换优先级为10，接着定义了4个A记录，不同域名指向了不同的IP地址。最后定义了一个www的别名，即访问linux.ixdba.net与访问www.ixdba.net是相同的。

下面接着分析一下反向区域数据文件60.168.192.zone的各个选项的含义：

```
[root@localhost named]#more /var/named/chroot/var/named/60.168.192.zone
$TTL    86400
60.168.192.in-addr.arpa.        IN      SOA     ns.ixdba.net. root.ixdba.net.(
                                        1997022700 ; Serial
                                        28800      ; Refresh
                                        14400      ; Retry
                                        3600000    ; Expire
                                        86400 )    ; Minimum
        IN      NS      ns.ixdba.net
136     IN      PTR     mail.ixdba.net.
```

135 IN PTR www.ixdba.net.

可以看出，基本结构与正向区域数据文件完全相同，只不过这里多出了一个PTR选项。PTR用来定义一个反向记录，也就是通过IP可以查到对应的域名信息。最后两行的第一列表示的是主机的IP地址，只不过省略了网络地址部分，如136对应的IP为192.168.60.136，同理，135对应的IP为192.168.60.135。

至此，DNS文件配置部分已经讲述完毕，从配置DNS的过程可以看出，DNS配置文件对格式要求非常严格，如果设置语句以空格键或tab 键开始的话，其设置被认为是一个“记录项”的内容，如果设置语句不是以空格键、Tab键开头，也不在SOA指定的括弧内，那么表示这个语句要定义一个新的“记录项”。因此，在修改配置文件时要特别小心。

3. 测试DNS配置

在对DNS文件的所有配置完成后，需要重启服务，以使配置生效，执行如下命令重启DNS服务：

```
[root@localhost named]#/etc/init.d/named restart
```

下面我们用nslookup命令对DNS解析情况进行测试。

```
[root@localhost ~]# nslookup
```

下面指定DNS服务器为本机，因为我们在DNS本机进行测试：

```
> server 127.0.0.1
Default server: 127.0.0.1
Address: 127.0.0.1#53
```

下面是测试测试A记录解析情况：

```
> www.ixdba.net
Server:         127.0.0.1
Address:        127.0.0.1#53
Name:   www.ixdba.net
Address: 192.168.60.135
> mail.ixdba.net
Server:         127.0.0.1
Address:        127.0.0.1#53
Name:   mail.ixdba.net
Address: 192.168.60.136
```

下面是测试CMANE记录解析情况：

```
> linux.ixdba.net
Server:         127.0.0.1
Address:        127.0.0.1#53
linux.ixdba.net canonical name = www.ixdba.net.
Name:   www.ixdba.net
Address: 192.168.60.135
```

下面是测试MX记录解析情况：

```
> set type=mx
> ixdba.net
Server:         127.0.0.1
Address:        127.0.0.1#53
ixdba.net       mail exchanger = 10 mail.ixdba.net.
```

下面是测试PTR记录解析情况：

```
> set type=PTR
> 192.168.60.135
Server:         127.0.0.1
Address:        127.0.0.1#53
135.60.168.192.in-addr.arpa      name = www.ixdba.net.
> 192.168.60.136
Server:         127.0.0.1
```

```
Address:        127.0.0.1#53
136.60.168.192.in-addr.arpa      name = mail.ixdba.net.
```

下面是测试NS记录解析情况：

```
> set type=ns
> ixdba.net
Server:         127.0.0.1
Address:        127.0.0.1#53
ixdba.net       nameserver = ns.ixdba.net.
```

从上面的输出可以看出，DNS都可以正确解析，说明我们上面的配置没有问题，DNS服务器已经可以工作了。