DNS配置实验小结

转载

weixin_33807284 于 2017-11-22 00:04:00 发布 2849 收藏 2
文章标签： 运维 网络
原文链接：https://yq.aliyun.com/articles/476938

DNS server是完成域名和IP之间正向、反向查询的一台服务提供者，由于互联网上的各种类型(例如Web、Mail、FTP等)的服务器地址都是由IP 构成，不方便记忆也不方便使用，因此需要使用一段方便记忆的文字(即域名)来翻译它进行使用，而DNS就是完成这个翻译过程的一个系统，DNS服务器就是这个翻译服务的提供者。
DNS命名用于Internet等TCP/IP网络中，通过用户名称查找计算机和服务。当用户在应用程序中输入DNS名称时，DNS服务可以将此名称解析为与之相关的其他信息，如IP地址。因为，你在上网时输入的网址，是通过域名解析系解析找到相对应的IP地址，这样才能上网。其实，域名的最终指向是IP。
配置DNS所需软件包列表：

软件包名称	作用
bind-libs	包含DNS的库文件
Bind	DNS服务器软件，安装此软件前需要安装libs
caching-nameserver	配置文件模板
bind-utils	DNS查询工具软件
bind-chroot	使DNS在chroot模式下运行，增强安全性（选择性安装）

在实验中，为了说明设置DNS的具体步骤，不安装bind-chroot这个文件包。只安装前面四个文件包就行了。
可以通过安装光盘安装文件包，也可以建立一个yum库来自动安装文件包。

好了，现在四个包文件都已经安装完毕了。需要说明的是DNS的主配置文件在/etc/named.conf中，区域配置文件在/var/named/目录下的一些文件，具体文件在后面的实验中会逐个提到，这里只是大概说明一下。
因为caching-nameserver包给我们提供了一些文件模板，在配置DNS文件时，我们可以参考这些模板。首先我们来编辑DNS的主配置文件。由于安装了caching-nameserver包，这就方便多了。在/etc/目录下，存在named.caching-nameserver.conf文件，这个就是主配置文件的模板文件。复制文件并命名为named.conf
【注：】在复制时，建议加上-p ，因为在运行DNS时，是named用户执行具体操作的。若文件权限不修改，在后面执行时会报错的，当然，复制完文件后，再来修改文件的属主属组权限也可以的。

接着要做的就是来编辑主配置文件了。在试验过程中以192.168.0.0/24为例子。
现在来简单定义一下主配置文件。

```
zone "localhost" IN {
type master;
file "localhost.zone";
};
```

type master;表示的意思是该DNS服务器为主DNS服务器。
file "localhost.zone"; 表示的意思是定义的该区域的区域文件名为localhost.zone
localhost 是定义本地正向区域文件；
0.0.127.in-addr.arpa 是定义本地反向区域文件；
.是定义根提示文件，该文件内容为全球公认的13台根DNS服务器；
liuht.com是定义本次试验的正向区域文件；
0.168.192.in-addr.arpa是定义本次试验的反向区域文件；

接下来要做的就是编辑区域配置文件了。区域配置文件在/var/named/目录下，由于在主配置文件中定义的本地区域文件和根提示文件已经存在（由caching-nameserver包提供并定义好）。我们要做的就是编辑本次试验的正向、反向区域配置文件。先来编辑正向文件吧？
在编辑区域文件时，切记要细心，因为一个小小的标点在这里也关闭重大。比如图中所示：
在该文件中存在着SOA 、NS 、A、 MX、CANME五种记录。
SOA 起始授权记录，该标签允许你配置此DNS区域的SOA记录。当DNS服务器加载DNS区域时，会通过SOA记录来决定此DNS区域的基本信息和主服务器，它还包括几个属性。具体如下：
主服务器：主服务器包含了此DNS区域的主DNS服务器的FQDN，此名字必须使用“.”结尾。
管理员邮箱：指定了管理此DNS区域的负责人的邮箱，当出现异常情况是给管理员发邮件，此名字必须使用“.”结尾（下图中就出现了这样的错误）。

serial 序列号：表示该区域文件的版本号（或者叫做修订号）。当区域中任何资源记录被修改时，此序列号也要手动修改。主要作用：当辅助DNS进行区域文件复制时，辅助DNS服务器查询主服务器上DNS区域的版本号，如果主服务器上DNS区域的版本号大于自己的版本号，则辅助DNS服务器向主DNS服务器发起区域复制。
refresh 更新时间间隔：表示辅助DNS服务器向主DNS服务器发起区域文件更新的时间间隔。当更新时间到期时，辅助DNS服务器从主DNS服务器上获取主DNS区域的SOA记录，然后和本地辅助DNS区域的SOA记录相比较，如果序列号值不相同则进行区域文件传输。默认情况下，刷新间隔为3小时。
retry 重试时间间隔：表示在辅助DNS服务器在向主DNS服务器发起区域文件更新请求失败的情况下，辅助DNS会等待一段时间之后，重新发起第二次更新请求，retry就是来定义这段时间的时间间隔。默认情况下，重试时间间隔为15分钟。
expiry 表示当主DNS服务器不在线时，辅助DNS服务器存活时间（继续工作的时间期限，若超过这个期限，则辅助DNS就不会提供地址解析服务）。默认值为一周。
minimum 表示当辅助DNS服务器得到负面回答（即主DNS服务器不在线）时，否定回答的有效时间段，默认为一天。

接下来要做的就是配置反向区域配置文件。在该文件中存在着SOA 、NS、 PTR记录。

到现在为止，我们的工作已经完成了一半了。我们还可以看一下本地区域配置文件内的内容，如图所示：
本地正向区域配置文件内容

本地反向区域配置文件

当所有文件都配置完成后，可以使用service named configtest命令来检查刚才的主配置文件是否存在语法错误。对于区域配置文件，可以用named-checkzone localhost /var/named/localhost命令来检查本地正向区域配置文件的语法。若要检查其他区域配置文件对应命令为 named-checkzone liuht.com /var/named/liuht.com.zone 后面两项参数与主配置文件的具体配置相对应。

当所有的语法检查都没有错误时，我们就可以在本机来测试一下DNS是否生效，常用的测试命令有 dig 和 host ，下面的几个图分别为dig和host的测试命令。
【注：】还要确保/etc/resolv.conf 文件中的nameserver 值为 192.168.0.84（实验主机IP）和127.0.0.1

除此之外，还可以用反向区域文件来测试一下。如图：

好了，到此为止，我们的DNS服务器就配置完成并测试通过了。但是再来考虑一下，在现实中，如果不怀好意的人来测试我们的DNS服务器的版本号信息，如图所示：

那么现在我们该怎么办呢？如果不显示版本信息，不就相对来说安全一些吗？行！接下来就不让别人获得DNS版本信息。
打开DNS主配置文件，/etc/named.conf 在options 项中添加
version “None of your business”;
如图所示：

再来执行一下刚才的命令 “ dig txt chaos version.bind.”
如图所示：

到此为止，本次DNS基本配置实验就结束了，有关DNS的高级配置，比如配置辅助DNS，子DNS的配置，会在后面的实验中逐步来演示。