任务背景

随着公司内部服务器的不断增多，现阶段公司所有服务器虽然都配置了主机名和IP的hosts解析，考虑到hosts文件的更新滞后，为了长远考虑，我们需要搭建一台DNS服务器，所有服务器都统一使用DNS服务来解析。

任务要求

1. 搭建一台DNS服务器
2. 配置域名的正反向解析

课程目标

- 了解DNS服务的作用
- 理解DNS服务的工作原理（理解）
- ==掌握DNS服务的正向解析配置(重点)==
- 能够搭建DNS的主从服务

涉及知识点

- DNS服务正向和反向解析配置（新知识点）

理论知识

DNS介绍

DNS（domain name system ） 域名管理系统

- 域名：

由特定的格式组成，用来表示互联网中==某一台计算机或者计算机组的名称==，能够是人更方便的访问互联网，而不用记住能够被机器直接读取的IP地址。

1. DNS的作用

- 域名的==正向解析==

  将主机域名转换为对应的IP 地址，以便网络程序能够通过主机域名访问到对应的服务器主机

  域名——>IP A记录

- 域名的==反向解析==

  将主机的IP地址转换为对应的域名，以便网络（服务）程序能够通过IP地址查询到主机的域名

  IP——>域名 PTR记录

2. DNS的结构

DNS结构图

根域 .

- 在整个 DNS 系统的最上方一定是 . (小数点) 这个 DNS 服务器 (称为 root)，也叫”根域“。
- 根域 <13台 全世界只有13台。1个为主根服务器，放置在美国。其余12个均为辅根服务器，其中9个放置在美国，欧洲2个，位于英国和瑞典，亚洲1个，位于日本。>

一级域名<顶级域|国家域>

com edu gov org cc io| cn uk us ru ja ko

二级域名

qq.com. baidu.com. google.com.

域名机构

收费<新网|万网> 老牌免费域名：TK顶级域名、TK域名DNS、TK域名商

http://www.freehao123.com

3. DNS工作原理

DNS解析原理

如果询问一次得到结果 递归查询 C-S 如果询问多次得到结果 迭代查询 S-S

一次递归 多次迭代

```
dig +trace www.baidu.com    追踪dns解析过程
dig @server www.baidu.com   正向解析查询
dig -x 192.168.0.1 @server  反向解析查询
dig +trace www.baidu.com    追踪一个域名解析过程
```

4. DNS服务软件

- DNS 的域名解析都是 **udp/53** . 主从之间的数据传输默认使用**tcp/53**
- DNS软件：

  Bind是一款开放源码的DNS服务器软件，Bind由美国加州大学Berkeley（伯克利）分校开发和维护的，全名为Berkeley Internet Name Domain它是目前世界上使用最为广泛的DNS服务器软件，支持各种unix平台和windows平台。BIND现在由互联网系统协会（Internet Systems Consortium）负责开发与维护。

任务实施

1. 环境介绍

```
以下主机属于web.cluster域
dns-server  10.1.1.1        提供主机名的正向和反向解析

web1    10.1.1.2            web1.web.cluster
web2    10.1.1.3            web2.web.cluster
web3    10.1.1.4            web3.web.cluster
app1    10.1.1.10       app1.web.cluster
app2    10.1.1.20       app2.web.cluster
app3    10.1.1.30       app3.web.cluster

搭建DNS服务，统一解析以上主机的域名(实现正反向解析)
```

2. 详细步骤

1. 关闭防火墙和selinux
2. 配置yum源
3. 软件三步曲
1）安装软件 bind
```
[root@dns-server ~]# yum -y install bind
```
2）确定是否成功安装
```
[root@dns-server ~]# rpm -q bind
bind-9.8.2-0.17.rc1.el6_4.6.x86_64
```
3）查看软件的文件列表
```
[root@dns-server ~]# rpm -ql bind
/etc/logrotate.d/named          //日志轮转文件
/etc/named                      //配置文件的主目录
/etc/named.conf                 主配置文件
/etc/named.rfc1912.zones         zone文件，定义域
/etc/rc.d/init.d/named           启动脚本

/usr/sbin/named                 二进制命令
/usr/sbin/named-checkconf       检查配置文件的命令    named.conf    named.rfc1912.zones
/usr/sbin/named-checkzone       检查区域文件的命令

/var/log/named.log              日志文件
/var/named                      数据文件的主目录
/var/named/data
/var/named/named.ca             根域服务器
/var/named/named.empty
/var/named/named.localhost      正向解析区域文件的模板
/var/named/named.loopback       反向解析区域文件的模板
/var/named/slaves               从dns服务器下载文件的默认路径
/var/run/named                  进程文件
```

4. 根据需求通过修改配置文件来完成服务的搭建
1） 修改主配置文件
备份配置文件：
```
[root@dns-server ~]# cp /etc/named.conf /etc/named.conf.bak
[root@dns-server ~]# cp /etc/named.rfc1912.zones /etc/named.rfc1912.zones.bak

[root@dns-server ~]# vim /etc/named.conf        定义监听端口和监听方式以及允许谁来查询
options {
    listen-on port 53 { 127.0.0.1;any; };       定义监听方式，any代表全网监听
    listen-on-v6 port 53 { ::1; };
    directory   "/var/named";
    dump-file   "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query     { localhost;any; };         允许任何人来查询
    recursion yes;
.....
};
....
```

2） 修改子配置文件
```
[root@dns-server ~]# vim /etc/named.rfc1912.zones       定义DNS服务器管理哪些域
```
在该文件的后面追加以下内容：

```
zone "web.cluster" IN {
        type master;
        file "web.cluster.zone";
        allow-update { none; };
};
zone "1.1.10.in-addr.arpa" IN {
        type master;
        file "10.1.1.zone";
        allow-update { none; };
};
```

3）需要在指定的数据目录里创建相应的zone文件

```
[root@dns-server ~]# cd /var/named/
[root@dns-server named]# cp -p named.localhost web.cluster.zone
[root@dns-server named]# cp -p named.loopback 10.1.1.zone
```

4）修改相应的区域文件（正向和反向）

```
[root@dns-server named]# cat web.cluster.zone
$TTL 1D
@       IN SOA  web.cluster. rname.invalid. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       NS      dns.web.cluster.
dns     A       10.1.1.1
web1    A       10.1.1.2
web2    A       10.1.1.3
web3    A       10.1.1.4
app1    A       10.1.1.10
app2    A       10.1.1.20
app3    A       10.1.1.30
```

注意：NS代表nameserver，前两行必须指定当前DNS服务的IP，abc.web.cluster.

```
[root@dns-server named]# cat 10.1.1.zone
$TTL 1D
@   IN SOA  web.cluster. rname.invalid. (
                    0   ; serial
                    1D  ; refresh
                    1H  ; retry
                    1W  ; expire
                    3H )    ; minimum
@   NS  dns.web.cluster.
2   PTR web1.web.cluster.
3   PTR web2.web.cluster.
4   PTR web3.web.cluster.
10  PTR app1.web.cluster.
20  PTR app2.web.cluster.

30  PTR app3.web.cluster.
```

注意：在反向记录文件中前面的A记录可以省略

5. 语法检测

```
[root@dns-server named]# named-checkconf /etc/named.conf
[root@dns-server named]# named-checkconf /etc/named.rfc1912.zones
[root@dns-server named]# named-checkzone web.cluster.zone web.cluster.zone
zone web.cluster.zone/IN: loaded serial 0
OK
[root@dns-server named]# named-checkzone 10.1.1.zone 10.1.1.zone
zone 10.1.1.zone/IN: loaded serial 0
OK
```

6. 启动服务，测试验证

```
[root@dns-server named]# service named start
Generating /etc/rndc.key:                                  [  OK  ]
Starting named:                                            [  OK  ]
[root@dns-server named]# netstat -nulp|grep named
udp        0      0 10.1.1.250:53               0.0.0.0:*
1550/named
udp        0      0 192.168.159.152:53          0.0.0.0:*
1550/named
udp        0      0 127.0.0.1:53                0.0.0.0:*
1550/named
udp        0      0 ::1:53                      :::*
 1550/named
```

```
[root@client ~]# echo nameserver 10.1.1.1 > /etc/resolv.conf
[root@client ~]# cat /etc/resolv.conf
nameserver 10.1.1.1
[root@client ~]# nslookup web1.web.cluster
Server:     10.1.1.1
Address:    10.1.1.1#53

Name:   web1.web.cluster
Address: 10.1.1.2
```

```
[root@client ~]# host web1.web.cluster
web1.web.cluster has address 10.1.1.2
[root@client ~]# host app3.web.cluster
app3.web.cluster has address 10.1.1.30
[root@client ~]# nslookup app3.web.cluster
Server:     10.1.1.1
Address:    10.1.1.1#53

Name:   app3.web.cluster
Address: 10.1.1.30

[root@client ~]# nslookup

> set type=a
```

```
> web2.web.cluster
Server:     10.1.1.1
Address:    10.1.1.1#53

Name:   web2.web.cluster
Address: 10.1.1.3
> app2.web.cluster
Server:     10.1.1.1
Address:    10.1.1.1#53

Name:   app2.web.cluster
Address: 10.1.1.20
> set type=ptr
> 10.1.1.20
Server:     10.1.1.1
Address:    10.1.1.1#53

20.1.1.10.in-addr.arpa  name = app2.web.cluster.
```

3. 课堂实战

搭建DNS服务器，要求管理test.org域，并且实现正向和反向解析，要求如下：

www.test.org 10.1.1.2/24

bbs.test.org 172.16.0.254/24

主配置文件：

```
vim /etc/named.conf
options {
        listen-on port 53 { 127.0.0.1; any; };      监听方式 ， any表示全网监听
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";    DNS缓存
        statistics-file "/var/named/data/named_stats.txt";      统计
        memstatistics-file "/var/named/data/named_mem_stats.txt";  内存统计
        allow-query     { localhost; any; };   允许哪些人可以查询；any代表任何人
        recursion yes;  是否递归

        dnssec-enable no;       dns安全扩展机制（签名认证）
        dnssec-validation no;
        dnssec-lookaside auto;

        /* Path to ISC DLV key */
        bindkeys-file "/etc/named.iscdlv.key";

        managed-keys-directory "/var/named/dynamic";
};
说明：
```

```
DNSSEC 域名系统安全，他是DNS的安全扩展协议
DLV DNSSEC 后备密钥
这些安全机制的设定，是为了保护DNS服务器与用户之间的数据安全，避免恶意数据对用户的欺骗

zone "." IN {      根域服务器
        type hint;
        file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

子配置文件：

```
vim /etc/named.rfc1912.zones
...
//定义正向域的模板
zone "localhost.localdomain" IN {
    type master;
    file "named.localhost";
    allow-update { none; };
};

//定义反向的模板
zone "1.0.0.127.in-addr.arpa" IN {
    type master;
    file "named.loopback";
    allow-update { none; };
};



# cat /var/named/named.localhost

$TTL  缓存的生存周期
@ = zonename = itcast.com  当前域
IN  互联网
SOA 开始授权
NS  dns服务端  nameserver
A   ipv4 正向
AAAA IPV6
CNAME 别名
MX  邮件交互记录  5 数字代表优先级 数字越小优先级越高


$TTL 1D
@       IN SOA  @ root.itcast.cn. (
                                        0       ; serial     更新序列号
                                        1D      ; refresh    更新间隔（从服务器下载数据）
                                        1H      ; retry      失败重试
                                        1W      ; expire     区域文件的过期时间

                                        3H )    ; minimum    缓存的最小生存周期
```

扩展总结

1. 多域搭建

```
需求：搭建一个DNS服务器，可以同时解析test.net和itcast.org域
具体要求：
ftp.test.net        10.1.1.3
www.itcast.org      10.1.1.2

环境：
dns-server  10.1.1.1

思路：
1. 关闭防火墙和selinux
2. 配置本地yum源
3. 软件三步曲
1）安装软件  bind
2）确认是否成功安装
3）查看软件的文件列表（配置文件、启动脚本、文档手册、二进制的命令）
4. 根据需求通过修改配置文件来完成服务的搭建（man 5）
5. 语法检测（不是必须）
6. 启动服务，测试验证    UDP/53   TCP/53

详细步骤：
1.略
2.略
3.略（在之前基础上完成）
4.根据需求修改配置文件
1）主配置文件   两个any不动
2）修改/etc/named.rfc1912.zones文件定义多个域

追加如下内容：
zone "test.net" IN {
        type master;
        file "test.net.zone";
        allow-update { none; };
};
zone "itcast.org" IN {
        type master;
        file "itcast.org.zone";
        allow-update { none; };
};

3）创建刚刚指定区域文件
[root@dns-server ~]# cd /var/named/
```

```
[root@dns-server named]# cp -p named.localhost test.net.zone
[root@dns-server named]# cp -p named.localhost itcast.org.zone

编写相应的文件:
[root@dns-server named]# cat test.net.zone
$TTL 1D
@   IN SOA  @ rname.invalid. (
                    0   ; serial
                    1D  ; refresh
                    1H  ; retry
                    1W  ; expire
                    3H )    ; minimum
    NS  dns.test.net.
dns A   10.1.1.1
ftp A   10.1.1.3

[root@dns-server named]# cat itcast.org.zone
$TTL 1D
@   IN SOA  @ rname.invalid. (
                    0   ; serial
                    1D  ; refresh
                    1H  ; retry
                    1W  ; expire
                    3H )    ; minimum
    NS  dns.itcast.org.
dns A   10.1.1.1
www A   10.1.1.2


5. 语法检测
[root@dns-server named]# named-checkconf /etc/named.conf
[root@dns-server named]# named-checkconf /etc/named.rfc1912.zones
[root@dns-server named]# named-checkzone itcast.org.zone itcast.org.zone
zone itcast.org.zone/IN: loaded serial 0
OK
[root@dns-server named]# named-checkzone test.net.zone test.net.zone
zone test.net.zone/IN: loaded serial 0
OK

6. 启动服务
[root@dns-server named]# service named start
Starting named:                                            [  OK  ]

7. 测试验证
```

2. 主从DNS搭建

环境:

ntp-server: 10.1.1.1

dns-master:10.1.1.1

dns-slave:10.1.1.2

client:10.1.1.3(Linux) 10.1.1.254(windows)

2.1 搭建时间同步服务器

==方法1：==

ntp （network time protocol） 端口：123

```
vim /etc/ntp.conf
增加以下行：
restrict 10.1.1.0 mask 255.255.255.0 nomodify notrap        //允许哪个网段来同步时间
注意：
当前时间同步服务器需要可以上外网，一搬需要等待3-5分钟。

客户端使用：
[root@dns-server named]# ntpdate 10.1.1.1
```

==方法2：==

```
需要安装xinetd软件：yum -y install xinetd

vim /etc/xinetd.d/time-stream
disable = no

vim /etc/xinetd.d/time-dgram
disable = no

service xinetd restart

[root@dns-server xinetd.d]# netstat -nltup|grep 37
tcp        0      0 :::37                       :::*                        LISTEN
 7196/xinetd
udp        0      0 :::37                       :::*
 7196/xinetd
```

总结：

ntpd时间同步服务依赖外网，客户端同步时需要等待几分钟

xinetd管理的时间同步服务，一般用于局域网中的时间同步

2.2 计划任务同步时间

```
dns-master: 10.1.1.1
dns-slave:10.1.1.2

在master和slave上都制定一个计划任务：
*/2 * * * * rdate -s 10.1.1.1 &>/dev/null
```

2.3 部署安装

环境：

master-dns：10.1.1.1

slave-dns：10.1.1.2

ntp-server：10.1.1.1

思路：

1. master和slave的系统时间保持一致
2. slave服务器上安装相应的软件（系统版本、软件版本高度保持一致）
3. 根据需求修改相应的配置文件（master和slave都应该去修改）
4. 主从同步的核心是**slave服务向master去下载**（同步）区域文件

步骤：

```
1.在master配置
vim /etc/named.conf
options {
        listen-on port 53 { 127.0.0.1; any;};
        allow-transfer {10.1.1.2;};     //允许哪个slave来同步下载区域文件
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query     { localhost;any; };
        recursion yes;
...
};


2. 在slave上配置
1)  安装bind软件
yum -y install bind

2）修改主配置文件
vim /etc/named.conf
打开全网监听，允许任何人来查询

3）修改子配置文件
vim /etc/named.rfc1912.zones
...
zone "test.net" IN {
        type slave;                 类型是slave
        file "slaves/test.net";  定义区域文件保存路径
         masters {10.1.1.1;};
};
zone "itcast.org" IN {
        type slave;
        file "slaves/itcast.org";
        masters {10.1.1.1;};
```

```
};
```

4）启动服务测试验证

```
[root@dns-slave named]# service named start
Generating /etc/rndc.key:                                  [  OK  ]
Starting named:                                            [  OK  ]
[root@dns-slave named]# ll slaves/
total 8
-rw-r--r-- 1 named named 336 Sep  3 11:53 itcast.org
-rw-r--r-- 1 named named 313 Sep  3 11:53 test.net

[root@ntp-server ~]# echo nameserver 10.1.1.250 > /etc/resolv.conf
[root@ntp-server ~]# echo nameserver 10.1.1.1 >> /etc/resolv.conf
[root@ntp-server ~]# cat /etc/resolv.conf
nameserver 10.1.1.250
nameserver 10.1.1.1
```

测试master修改区域文件后，slave自动下载：

注意：

master通过将版本号修改大进而告知slave该文件更新，就会自动同步

比如：

```
$TTL 1D
@   IN SOA  @ rname.invalid. (
                    2018090308  ; serial        //第一次为2018090301，版本号更新变大说明文件有更新
                    1M  ; refresh
                    1M  ; retry
                    1M  ; expire
                    3M )    ; minimum
    NS  dns.test.net.
dns A   10.1.1.250
ftp A   172.16.0.10
bbs A   10.1.1.222
aaa CNAME   ftp.test.net.
bbb A   172.16.0.222
```

master修改完后需要重启服务。

slave端测试：

```
[root@dns-slave slaves]# ll
total 8
-rw-r--r-- 1 named named 336 Sep  3 15:15 itcast.org
-rw-r--r-- 1 named named 378 Sep  3 17:16 test.net
```