Configure linux dns slave server step by step guide example and implementation

A DNS server, or name server, is used to resolve an IP address to a hostname or vice versa. You can set up four different types of DNS servers:

- A **master DNS server for your domain(s),** which stores authoritative records for your domain.
- A **slave DNS server,** which relies on a master DNS server for data.
- A **caching-only DNS server,** which stores recent requests like a proxy server. It otherwise refers to other DNS servers.
- A **forwarding-only DNS server,** which refers all requests to other DNS servers.

In our pervious article we have configured **Master dns server**. In this tutorial we will extended pervious article and configure **salve dns server.** This article is the second part of How to configure dns server step by step guide. We suggest you to review pervious article before starting it.

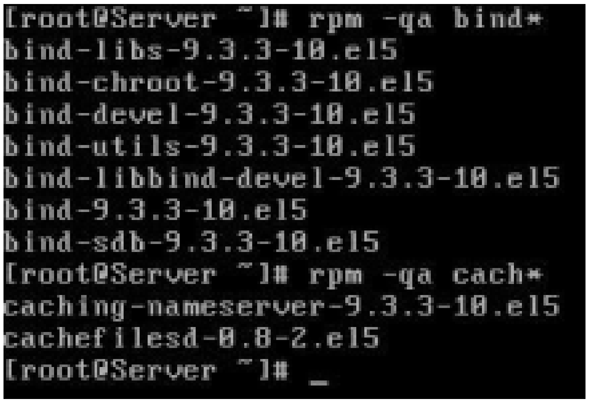
Configure dns slave server

For this example we are using three systems one linux server one linux clients and one window clients.

We have configured master DNS server with ip address of **192.168.0.254** and hostname **server.example.com** on linux server. Now we will configure **slave DNS server** on linux clients

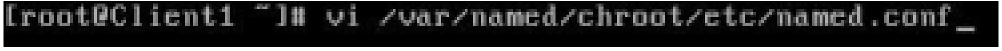
To configure slave DNS server go on client1 system.

First test connectivity from dns server by ping commands and check necessary rpm. **bind** and **caching-nameserver** rpm is required to configure dns. check them for install if not found install them.



```
[root@Server ~]# rpm -qa bind*
bind-libs-9.3.3-10.el5
bind-chroot-9.3.3-10.el5
bind-devel-9.3.3-10.el5
bind-utils-9.3.3-10.el5
bind-libbind-devel-9.3.3-10.el5
bind-9.3.3-10.el5
bind-sdb-9.3.3-10.el5
[root@Server ~]# rpm -qa cach*
caching-nameserver-9.3.3-10.el5
cachefilesd-0.8-2.el5
[root@Server ~]# _
```

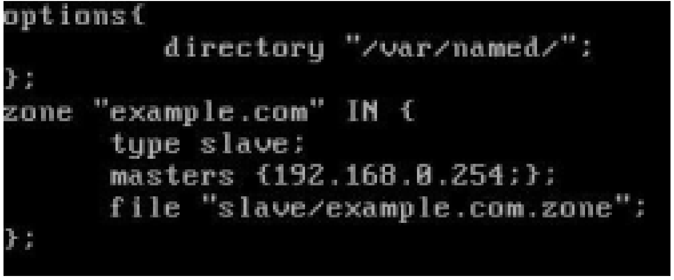
set hostname to **client1** and ip address to **192.168.0.1** And create a new **named.conf** file



```
[root@Client1 ~]# vi /var/named/chroot/etc/named.conf_
```

We are using bind's **chroot** features so all our necessary files will be located in chroot directory. Set directory location to **/var/named.** As we are configuring **slave server** so we need not to define the location of zone database files. Zone database file can be created and modified only on master server. A **slave server** only copied it's from master server.

Do editing exactly as shown here in image in **named.conf**



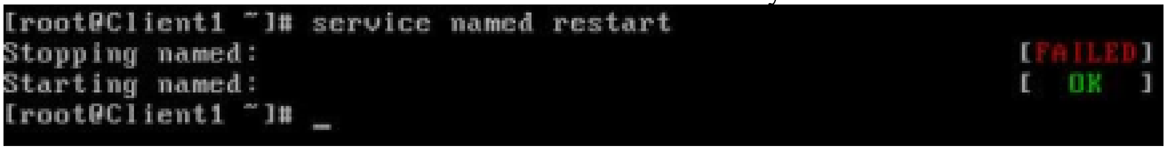
```
options{
        directory "/var/named/";
};
zone "example.com" IN {
      type slave;
      masters {192.168.0.254;};
      file "slave/example.com.zone";
};
```

save this file with **:wq** and exit

If you cannot create this file manually then download this pre configured file and copy to **/var/named/chroot/etc/named.conf**

To download do right click here and choose save link As..
named.conf

Now restart the named service. It should be start without any error.

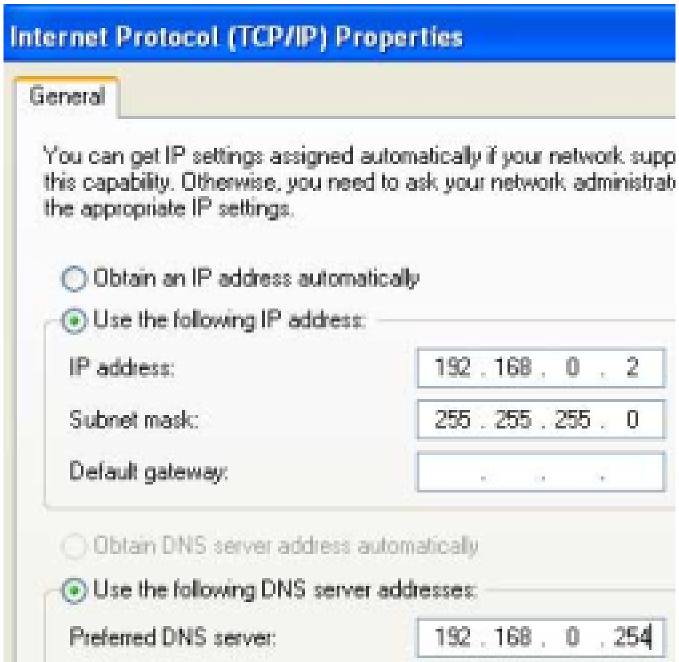


```
[root@Client1 ~]# service named restart
Stopping named:                                                  [FAILED]
Starting named:                                                  [  OK  ]
[root@Client1 ~]# _
```

Congratulation you have configured both Master and client DNS server. Now we will configure dns client and test it with dns server.

Configure Window DNS Client

Now go on windows xp system and test connectivity from DNS server. And set **DNS ip address** in LAN card properties.



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supp
this capability. Otherwise, you need to ask your network administrat
the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 0 . 2

Subnet mask: 255 . 255 . 255 . 0

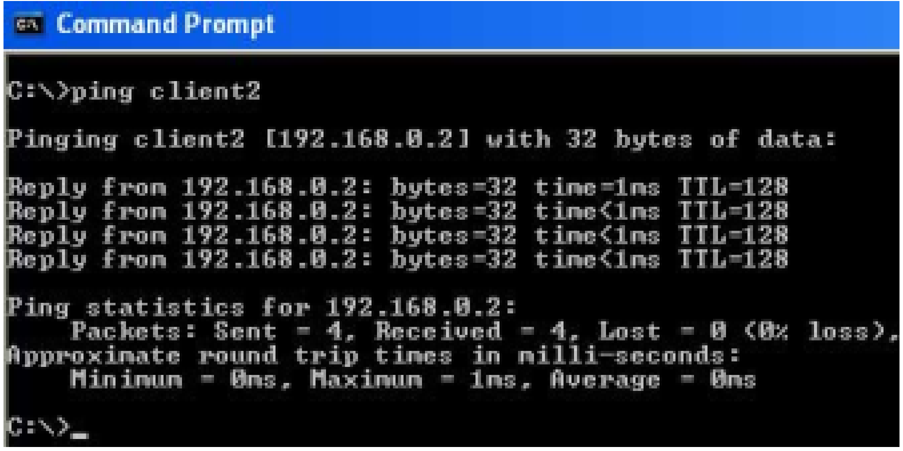
Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

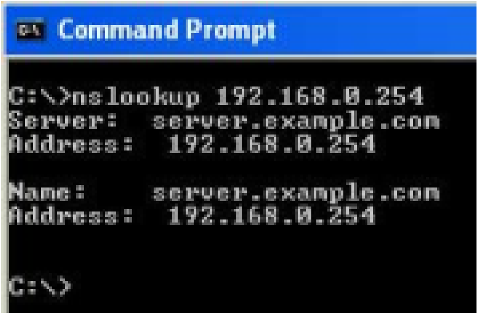
Preferred DNS server: 192 . 168 . 0 . 254

Now go on **commands prompt** and ping from other client by name to test **dns**.



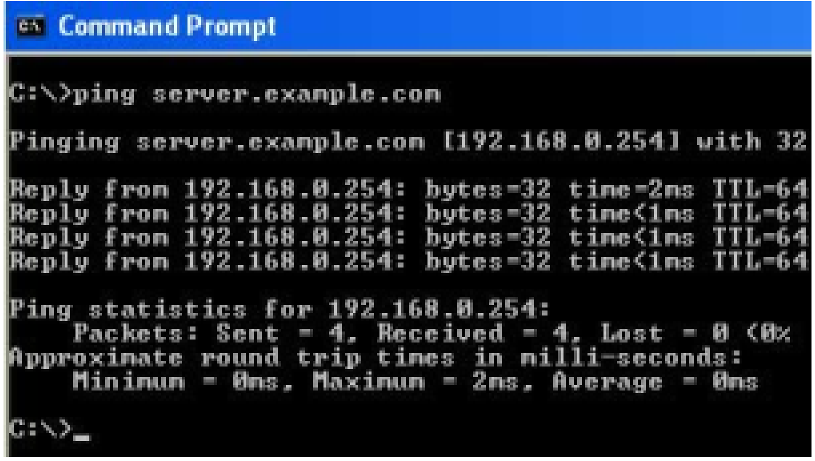
```
C:\>ping client2

Pinging client2 [192.168.0.2] with 32 bytes of data:

Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>_
```

Alternately You can also verify **DNS server** by **nslookup** command



```
C:\>nslookup 192.168.0.254
Server:  server.example.com
Address:  192.168.0.254

Name:    server.example.com
Address:  192.168.0.254


C:\>
```

Test also by **pinging server** from name



```
C:\>ping server.example.com

Pinging server.example.com [192.168.0.254] with 32

Reply from 192.168.0.254: bytes=32 time=2ms TTL=64
Reply from 192.168.0.254: bytes=32 time<1ms TTL=64
Reply from 192.168.0.254: bytes=32 time<1ms TTL=64
Reply from 192.168.0.254: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0%
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\>_
```

Configure Linux DNS clients

RHCE Exam question

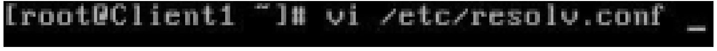
Dig Server.example.com, Resolve to successfully through DNS Where DNS server is 192.168.0.254.

RHCE Exam question2

Your System is configured in 192.168.0.0/24 Network and your nameserver is 192.168.0.254. Make successfully resolve to server.example.com.

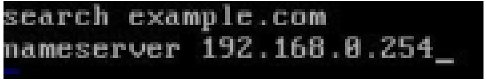
On command line interface you don't have any options to set DNS ip in network configuration window. IP of DNS server can be set from **/etc/resolv.conf** file. Each nameserver line represents a DNS server, and the search line specifies domain names to try if only the first part of a hostname is used. For example, if just the name client1 is used as a hostname, **client1. example.com** will also be tried if the /etc/resolv.conf file is configured as shown in image below on the system.

To set DNS ip open **/etc/resolv.conf** file



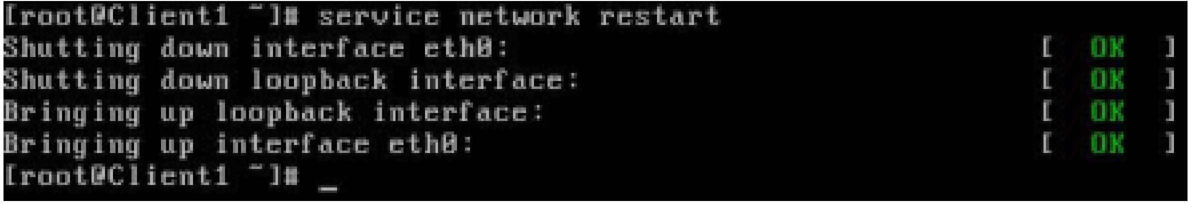
```
[root@Client1 ~]# vi /etc/resolv.conf _
```

set **nameserver** ip to **192.168.0.254** and **search** option to **example.com**



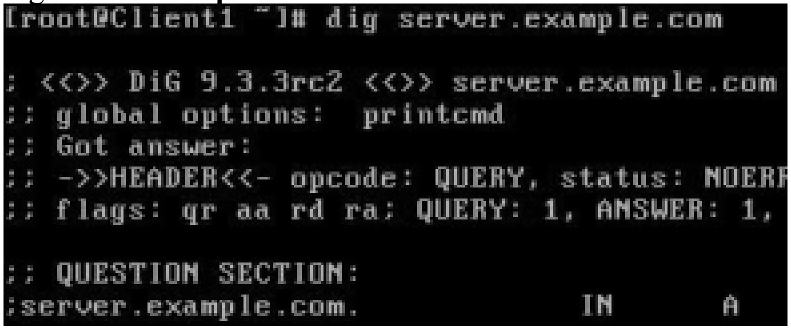
```
search example.com
nameserver 192.168.0.254_
```

After saving **/etc/resolv.conf** file restart the network service



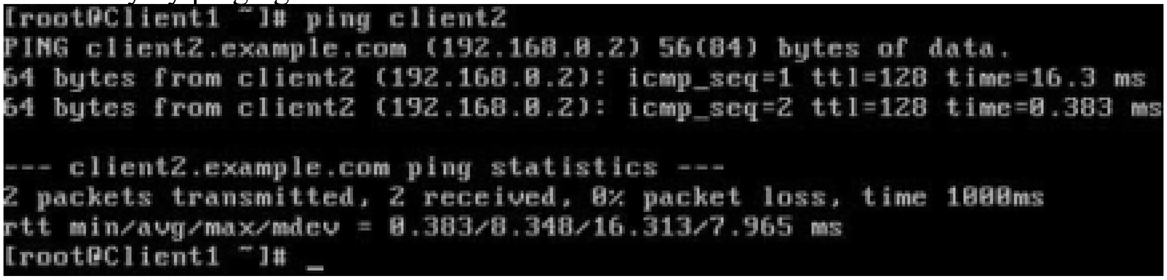
```
[root@Client1 ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
[root@Client1 ~]# _
```

dig **server.example.com** to test dns server



```
[root@Client1 ~]# dig server.example.com

; <<>> DiG 9.3.3rc2 <<>> server.example.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERR
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1,

;; QUESTION SECTION:
;server.example.com.            IN      A
```

now verify by pinging to other client from name



```
[root@Client1 ~]# ping client2
PING client2.example.com (192.168.0.2) 56(84) bytes of data.
64 bytes from client2 (192.168.0.2): icmp_seq=1 ttl=128 time=16.3 ms
64 bytes from client2 (192.168.0.2): icmp_seq=2 ttl=128 time=0.383 ms

--- client2.example.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.383/8.348/16.313/7.965 ms
[root@Client1 ~]# _
```