# netkit lab

## load balancer – dns

<table>
<thead>
<tr>
<th>Version</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Massimo Rimondini</td>
</tr>
<tr>
<td><strong>E-mail</strong></td>
<td><a href="mailto:contact@netkit.org">contact@netkit.org</a></td>
</tr>
<tr>
<td><strong>Web</strong></td>
<td><a href="http://www.netkit.org/">http://www.netkit.org/</a></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>A lab showing how to perform simple load balancing on a set of web servers using the DNS</td>
</tr>
</tbody>
</table>
copyright notice

- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright.

- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.

- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.

- Information contained in this material cannot be used within network design projects or other products of any kind.

- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.

- The authors assume no responsibility about this material and provide this material “as is”, with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.

- This copyright notice must always be redistributed together with the material, or its portions.
lab topology
lab topology

US server farm

server3

authority for web.com

ns1

authority for example.us

root-ns

EU server farm

daughter for example.us

daughter for test.eu

authority for web.com
lab topology

US server farm

server1

server2

server3

WEB

WEB

WEB

ns

server4

server5

WEB

WEB

root-ns

NS

E

US server farm

EU server farm

A

client1

ns1

r1

B

client2

ns2

r2

C

D

E
lab description – web servers

- server\{1,2,3,4,5\} are all web servers running apache2
- each server hosts a single default HTML page (in /var/www/index.html) with different contents, to easily distinguish one server from the others
- server3 also runs bind, because it is the authority for zone web.com
lab description – routers

- to make the setup simpler, static routes are used on all devices, including routers

```bash
ifconfig eth0 100.0.0.1 netmask 255.255.255.0 up
ifconfig eth1 150.0.0.1 netmask 255.255.255.252 up
ifconfig eth2 200.0.0.1 netmask 255.255.255.0 up
route add -net 10.0.0.0/24 gw 100.0.0.2
route add -net 20.0.0.0/24 gw 200.0.0.2
```

lab description – clients

- have a text-based web browser (links)
- have the system-wide resolver configured to point to the local name server

```plaintext
client1's /etc/resolv.conf
nameserver 10.0.0.2
domain example.us
client2's /etc/resolv.conf
nameserver 20.0.0.2
domain test.eu
```
lab description – name servers

- **root-ns** is the root name server
- there are no authoritative servers for zones “com.”, “us.”, and “eu.”
- **ns1** is the authority for **example.us**.
  - try pinging **client.example.us** and **ns.example.us** from the clients
- **ns2** is the authority for **test.eu**.
  - try pinging **client.test.eu** and **ns.test.eu** from the clients
- **server3** is the authority for **web.com**.
  - **www.web.com** is the name behind which the farm offers the web service
  - TTLs are zeroed to appreciate load balancing
lab description – name servers

- different load balancing mechanisms
  - round robin
  - location-based
  - random

- all simultaneously operating in the same lab
lab description – name servers

- there are multiple A records for www.web.com, one for each server in the farm

```
server3's /etc/bind/db.web.com-us

$TTL 0 ; do not cache, so that we can appreciate
; load balancing
@ IN SOA web.com. root.localhost. (  
 2 ; Serial
 604800 ; Refresh
 86400 ; Retry
2419200 ; Expire
 604800 ) ; Negative Cache TTL

; @
@ IN NS ns.web.com.
ns IN A 100.0.0.5

www IN A 100.0.0.3
www IN A 100.0.0.4
www IN A 100.0.0.5
```

round robin
lab description – name servers

- there are multiple A records for `www.web.com`, one for each server in the farm
- these records are all returned to the client when it performs a query

```
client1:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> www.web.com +noall +answer
;; global options:  printcmd
www.web.com.            0       IN      A       100.0.0.5
www.web.com.            0       IN      A       100.0.0.3
www.web.com.            0       IN      A       100.0.0.4
```

round robin
lab description – name servers

- there are multiple A records for www.web.com, one for each server in the farm
- these records are all returned to the client when it performs a query
- user applications (e.g., web browsers) usually only consider the first returned record
- by default, bind “rotates” returned records in a round robin fashion

round robin load balancing
lab description – name servers

round robin load balancing

client2:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> www.web.com +noall +answer
;; global options:  printcmd
www.web.com.          0      IN      A       200.0.0.6
www.web.com.          0      IN      A       200.0.0.7
client2:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> www.web.com +noall +answer
;; global options:  printcmd
www.web.com.          0      IN      A       200.0.0.7
www.web.com.          0      IN      A       200.0.0.6
client2:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> www.web.com +noall +answer
;; global options:  printcmd
www.web.com.          0      IN      A       200.0.0.6
www.web.com.          0      IN      A       200.0.0.7
client2:~# dig www.web.com +noall +answer
lab description – name servers

round robin load balancing

- start
  - links http://www.web.com/
on client2 and issue “disconnect (ctrl+S) & reload (ctrl+R)” multiple times
- you should see pages coming from different servers
lab description – name servers

round robin load balancing

- the `count_server_replies.sh` script on the clients issues 100 HTTP requests to `www.web.com` and counts the number of times the reply came from each server

```
client2:~# ./count_server_replies.sh
Sending 100 requests to www.web.com...
  50 replies received from server 4
  50 replies received from server 5
```
lab description – name servers

- **server3** is the sole authority for **www.web.com**, but different countries are offered different views of the DNS database, to improve load balancing

```
server3's /etc/bind/named.conf

view "US" {
    match-clients { 10.0.0.0/24; };
    ...
    zone "web.com" {
        type master;
        file "/etc/bind/db.web.com-us";
    };
    rrset-order { order random; };
};

view "Europe" {
    match-clients { 20.0.0.0/24; };
    ...
    zone "web.com" {
        type master;
        file "/etc/bind/db.web.com-eu";
    };
    rrset-order { order cyclic; };
};
```
lab description – name servers

- server3 is the sole authority for www.web.com, but different countries are offered different views of the DNS database, to improve load balancing.

location-based load balancing
lab description – name servers

location-based load balancing

```
client1:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<< www.web.com +noall +answer
;; global options: printcmd
www.web.com.            0       IN      A       100.0.0.4
www.web.com.            0       IN      A       100.0.0.5
www.web.com.            0       IN      A       100.0.0.3

client2:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<< www.web.com +noall +answer
;; global options: printcmd
www.web.com.            0       IN      A       200.0.0.7
www.web.com.            0       IN      A       200.0.0.6
```
lab description – name servers

location-based load balancing

- start

  links http://www.web.com/
on client1 and client2 and see that
pages are always obtained from different
servers
lab description – name servers

location-based load balancing

- also check with count_server_replies.sh

```bash
client1:~# ./count_server_replies.sh
Sending 100 requests to www.web.com...
  33 replies received from server 1
  33 replies received from server 2
  34 replies received from server 3

client2:~# ./count_server_replies.sh
Sending 100 requests to www.web.com...
  50 replies received from server 4
  50 replies received from server 5
```
lab description – name servers

- different orderings are used when returning records to clients
  - default is round robin (cyclic)

server3’s `/etc/bind/named.conf`

```conf
view "US" {
    match-clients { 10.0.0.0/24; }
    ... 
    zone "web.com" {
        type master;
        file "/etc/bind/db.web.com-us";
    }
    rrset-order { order random; };
}

view "Europe" {
    match-clients { 20.0.0.0/24; }
    ... 
    zone "web.com" {
        type master;
        file "/etc/bind/db.web.com-eu";
    }
    rrset-order { order cyclic; };
}
```
Different orderings are used when returning records to clients:
- Default is round robin (cyclic)

```
server3's /etc/bind/named.conf
view "US" {
    match-clients { 10.0.0.0/24; };
    ...
    zone "web.com" {
        type master;
        file "/etc/bind/db.web.com-us";
    };
    rrset-order { order random; };
};
```

So, US clients are supposed to see records returned in random order...
lab description – name servers

- so, US clients are supposed to see records returned in random order...

random load balancing
lab description – name servers

random load balancing

client1:~# dig www.web.com +noall +answer

; <<< DiG 9.5.0-P2 <<< www.web.com +noall +answer
;; global options: printcmd
www.web.com. 0 IN A 100.0.0.3
www.web.com. 0 IN A 100.0.0.4
www.web.com. 0 IN A 100.0.0.5

client1:~# dig www.web.com +noall +answer

; <<< DiG 9.5.0-P2 <<< www.web.com +noall +answer
;; global options: printcmd
www.web.com. 0 IN A 100.0.0.4
www.web.com. 0 IN A 100.0.0.5
www.web.com. 0 IN A 100.0.0.3

client1:~# dig www.web.com +noall +answer

; <<< DiG 9.5.0-P2 <<< www.web.com +noall +answer
;; global options: printcmd
www.web.com. 0 IN A 100.0.0.5
www.web.com. 0 IN A 100.0.0.3
www.web.com. 0 IN A 100.0.0.4
lab description – name servers

random load balancing

client1:

client1:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> www.web.com +noall +answer
;; global options: printcmd
www.web.com. 0 IN A 100.0.0.3
www.web.com. 0 IN A 100.0.0.4
www.web.com. 0 IN A 100.0.0.5

client1:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> www.web.com +noall +answer
;; global options: printcmd
www.web.com. 0 IN A 100.0.0.4
www.web.com. 0 IN A 100.0.0.5
www.web.com. 0 IN A 100.0.0.3

client1:~# dig www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> www.web.com +noall +answer
;; global options: printcmd
www.web.com. 0 IN A 100.0.0.5
www.web.com. 0 IN A 100.0.0.3
www.web.com. 0 IN A 100.0.0.4

but this is round robin!
beware of address sorting

- BIND 4 servers, by default, sort addresses if [...] the host that sent the query to the name server shares a network with the name server host. [...] When BIND starts up, it finds all the interface addresses of the host it's running on. [...] to create the default sort list. When a query is received, BIND checks whether the sender's address is on a network in the default sort list. If it is, then the query is local and BIND sorts the addresses in the response.

beware of address sorting

- address sorting happens at 3 places:
  1. server (*rrset-order*)
  2. client (default behavior or explicit *sortlist*)
  3. end user’s resolver
lab description – name servers

random load balancing

■ so what?

■ you can still appreciate random load balancing by directly querying `ns.web.com (100.0.0.5)`

```
client1:~# dig @ns.web.com www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> @ns.web.com www.web.com +noall +answer
; (1 server found)
;; global options:  printcmd
www.web.com.            0       IN      A       100.0.0.4
www.web.com.            0       IN      A       100.0.0.3
www.web.com.            0       IN      A       100.0.0.5

client1:~# dig @ns.web.com www.web.com +noall +answer

; <<>> DiG 9.5.0-P2 <<>> @ns.web.com www.web.com +noall +answer
; (1 server found)
;; global options:  printcmd
www.web.com.            0       IN      A       100.0.0.3
www.web.com.            0       IN      A       100.0.0.4
www.web.com.            0       IN      A       100.0.0.5
```
Start

Links http://www.web.com/ on client1 and client2 and see the effects of

- round robin
- location-based
- random

Load balancing on the choice of the machine that serves the request.
experiments

- change client1’s resolver configuration to use 100.0.0.5 as name server
- start
  links http://www.web.com/
  on client1, issue “disconnect (ctrl+S) & reload (ctrl+R)” multiple times, and appreciate the effect of random load balancing
- ...or use count_server_replies.sh to do the same
questions

answering these proposed questions requires interaction with the lab and investigation in the bind documentation.
questions

- for the round robin case, does bind rotate records independently for each client?
  [no: consider that that would mean keeping a state for each client; check it]

- does changing the TTL of resource records “kill” load balancing?
  [no, because local name servers keep rotating records when returning them to client1 and client2, even though none of the TTLs has expired; check it]

- consider a set of multiple resource records for the same resource (e.g., www.web.com): what TTL is returned to clients if different records have different TTLs?
  [the TTL of the first record in the set; check it]
(more advanced) questions

- try querying **100.0.0.5 (server3)** for **www.web.com** from a machine that is outside the customer networks, both recursively and iteratively: does the server reply?
  
  [no, because the client’s address does not fall into any of the subnets specified in **match-clients** statements]

- how could you reconfigure **server3** to answer queries for **www.web.com** from any hosts?
  
  [by setting one of the existing **match-clients** statements to match 0.0.0.0/0 or by defining an additional view that matches 0.0.0.0/0; note that bind does not tolerate any zone statements outside views if at least one **view** is defined; moreover, bind always replies according to the first matching view, considering them in the order in which they are specified in **named.conf**]
([even] more advanced) questions

- try querying 10.0.0.2 (ns1) for www.web.com from any location in the network (particularly from the EU), both recursively and iteratively: does the server reply?

  [only when the query comes from a client in the same 10.0.0.0/24 subnet; although by default bind accepts queries from any clients, if the query comes from a different subnet than the server’s and it implies an attempt to access the server’s cache, then bind refuses it; in general, queries that require “climbing up” the name hierarchy imply a cache access attempt to “avoid the climbing”; this behavior can be overridden by using the allow-query-cache configuration option, that defaults to {localnets; localhost;};]

- try querying 10.0.0.2 (ns1) for client.example.us from any location in the network (particularly from the EU), both recursively and iteratively: does the server reply?

  [yes, because ns1 is the authority for client.example.us and therefore this query does not imply any cache access attempts]