

Otto-Friedrich-University of Bamberg

Professorship for Computer Science,
Communication Services, Telecommunication
Systems and Computer Networks



Foundations of Internet Communication

Assignment 3 Domain Name System (DNS) and Load Balancing

Submitted by:

Group X

Moktahid Al Faisal
Abdullah Al Mosabbir
Mohammed Mehedi Hasan
Kazi Sayef Shawgat
Sheikh Jumon Ahmed

Supervisor: Prof. Dr. Udo Krieger

Bamberg, June 7, 2020
Summer Term 2020

Contents

1	The Domain Information Groper (dig)	1
2	DNS Configuration with CoreDNS	2
3	Load Balancing with Traefik	8

List of Figures

1	authoritative DNS server for ru	1
2	Address of the Internet DNS root servers	1
3	Nameservers for domain uni-bamberg.de	1
4	lab.conf file of Table 1	2
5	Startup files of web_amy, web_penny, web_bernadette	2
6	Startup files of web_amy, web_penny, web_bernadette	3
7	Startup files of web_1	3
8	Startup files of pc_1 and pc_2	3
9	Startup files of dns_root, dns_lb	3
10	Static routes to the topology	4
11	Screenshot of dns setup	4
12	Screenshot of resolv.conf and curl gik.de	5
13	Screenshot of authoritative server	5
14	Screenshot of dns_lb setup	6
15	Screenshot of modified dns_root setup	6
16	Screenshot of configured core dns and load balancing	7
17	Screenshot of dig on gik.org	8
18	websheldon.startup file	8
19	webleonard.startup file	8
20	webhoward.startup file	8
21	trafiklbStartup.startup file	9
22	80 Listen Port	9
23	File Provider for new webserver	9
24	update dns root A record	10

25	dig confirmation	10
26	Update router r1 to adjust traefik	10
27	Weighted round robin load balancer configuration	11
28	Test result of load balancer	11

1 The Domain Information Groper (dig)

1. Determine the authoritative DNS server for the top level domain *ru*

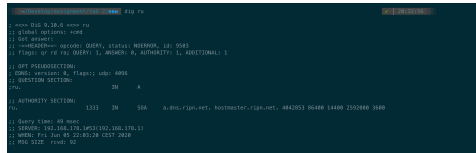


Figure 1: authoritative DNS server for ru

- Determine the addresses of the Internet DNS root servers

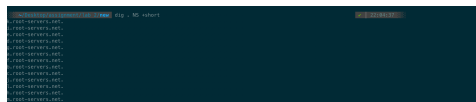


Figure 2: Address of the Internet DNS root servers

3. Run `dig` to display the nameservers for the domain `uni-bamberg.de`

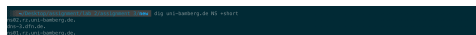
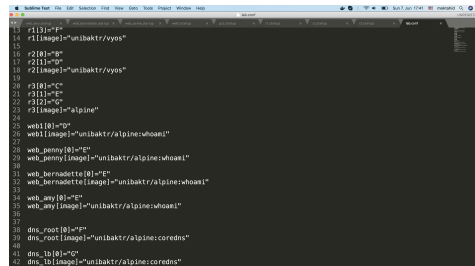


Figure 3: Nameservers for domain uni-bamberg.de

2 DNS Configuration with CoreDNS

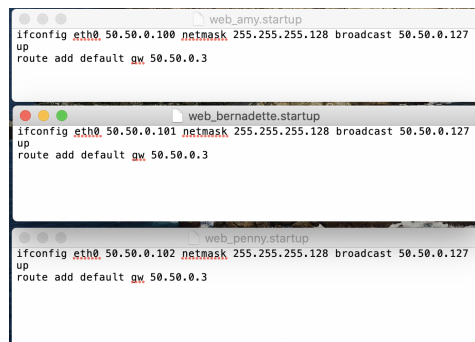
1. Followings is the screenshot of lab.conf file:



```
13 r1[1]="r1"
14 r1[image]="unibaktr/ryes"
15
16 r2[1]="r2"
17 r2[1]="r2"
18 r2[image]="unibaktr/ryes"
19
20 r3[1]="r3"
21 r3[1]="r3"
22 r3[2]="r3"
23 r3[image]="alpine"
24
25 web1[1]="r1"
26 web1[image]="unibaktr/alpine:whoami"
27
28 web_penny[1]="r1"
29 web_penny[image]="unibaktr/alpine:whoami"
30
31 web_bernadette[1]="r1"
32 web_bernadette[image]="unibaktr/alpine:whoami"
33
34 web_amy[1]="r1"
35 web_amy[image]="unibaktr/alpine:whoami"
36
37 dns_root[1]="r1"
38 dns_root[image]="unibaktr/alpine:coredns"
39
40 dns_lb[1]="r1"
41 dns_lb[image]="unibaktr/alpine:coredns"
```

Figure 4: lab.conf file of Table 1

2. Followings are the screenshots of startup files file:

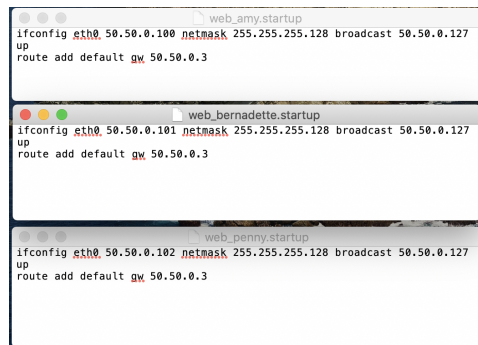


```
web_amy.startup
ifconfig eth0 50.50.0.100 netmask 255.255.255.128 broadcast 50.50.0.127
up
route add default gw 50.50.0.3

web_bernadette.startup
ifconfig eth0 50.50.0.101 netmask 255.255.255.128 broadcast 50.50.0.127
up
route add default gw 50.50.0.3

web_penny.startup
ifconfig eth0 50.50.0.102 netmask 255.255.255.128 broadcast 50.50.0.127
up
route add default gw 50.50.0.3
```

Figure 5: Startup files of web_amy, web_penny, web_bernadette

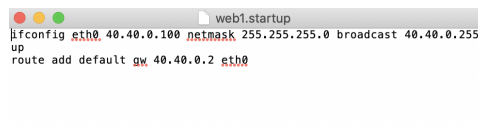


```
web_amy.startup
ifconfig eth0 50.50.0.100 netmask 255.255.255.128 broadcast 50.50.0.127
up
route add default gw 50.50.0.3

web_bernadette.startup
ifconfig eth0 50.50.0.101 netmask 255.255.255.128 broadcast 50.50.0.127
up
route add default gw 50.50.0.3

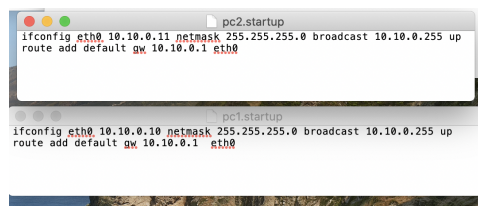
web_penny.startup
ifconfig eth0 50.50.0.102 netmask 255.255.255.128 broadcast 50.50.0.127
up
route add default gw 50.50.0.3
```

Figure 6: Startup files of web_amy, web_penny, web_bernadette



```
web1.startup
ifconfig eth0 40.40.0.100 netmask 255.255.255.0 broadcast 40.40.0.255
up
route add default gw 40.40.0.2 eth0
```

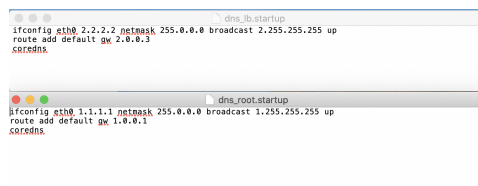
Figure 7: Startup files of web_1



```
pc2.startup
ifconfig eth0 10.10.0.11 netmask 255.255.255.0 broadcast 10.10.0.255 up
route add default gw 10.10.0.1 eth0

pc1.startup
ifconfig eth0 10.10.0.10 netmask 255.255.255.0 broadcast 10.10.0.255 up
route add default gw 10.10.0.1 eth0
```

Figure 8: Startup files of pc_1 and pc_2

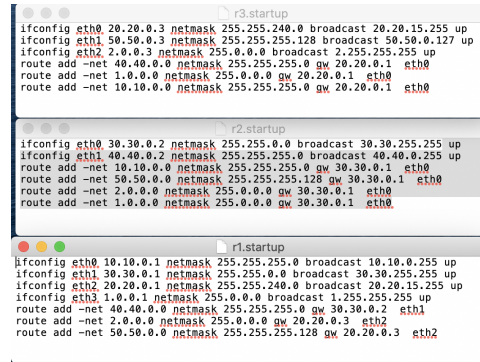


```
dns_lb.startup
ifconfig eth0 2.2.2.2 netmask 255.0.0.0 broadcast 2.255.255.255 up
route add default gw 2.0.0.3
soredns

dns_root.startup
ifconfig eth0 1.1.1.1 netmask 255.0.0.0 broadcast 1.255.255.255 up
route add default gw 1.0.0.1
soredns
```

Figure 9: Startup files of dns_root, dns_lb

3. Static routes to the topology :



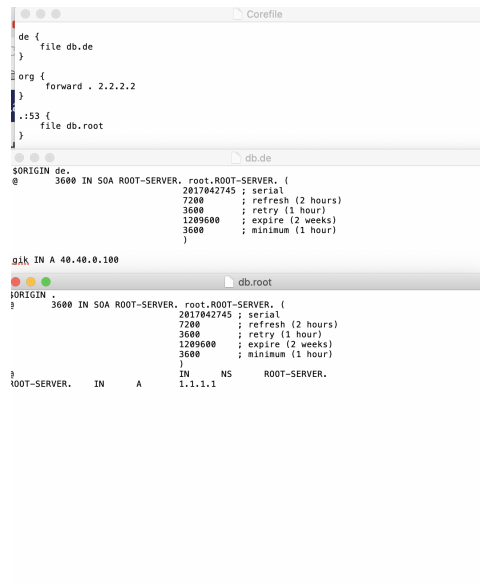
```
r3.startup
ifconfig eth0 20.20.0.3 netmask 255.255.240.0 broadcast 20.20.15.255 up
ifconfig eth1 50.50.0.3 netmask 255.255.255.128 broadcast 50.50.0.127 up
ifconfig eth2 2.0.0.3 netmask 255.0.0.0 broadcast 2.255.255.255 up
route add -net 40.40.0.0 netmask 255.255.255.0 gw 20.20.0.1 eth0
route add -net 1.0.0.0 netmask 255.0.0.0 gw 20.20.0.1 eth0
route add -net 10.10.0.0 netmask 255.255.255.0 gw 20.20.0.1 eth0

r2.startup
ifconfig eth0 30.30.0.2 netmask 255.255.0.0 broadcast 30.30.255.255 up
ifconfig eth1 40.40.0.2 netmask 255.255.255.0 broadcast 40.40.0.255 up
route add -net 10.10.0.0 netmask 255.255.255.0 gw 30.30.0.1 eth0
route add -net 50.50.0.0 netmask 255.255.255.128 gw 30.30.0.1 eth0
route add -net 2.0.0.0 netmask 255.0.0.0 gw 30.30.0.1 eth0
route add -net 1.0.0.0 netmask 255.0.0.0 gw 30.30.0.1 eth0

r1.startup
ifconfig eth0 10.10.0.1 netmask 255.255.255.0 broadcast 10.10.0.255 up
ifconfig eth1 30.30.0.1 netmask 255.255.0.0 broadcast 30.30.255.255 up
ifconfig eth2 20.20.0.1 netmask 255.255.240.0 broadcast 20.20.15.255 up
ifconfig eth3 1.0.0.1 netmask 255.0.0.0 broadcast 1.255.255.255 up
route add -net 40.40.0.0 netmask 255.255.255.0 gw 30.30.0.2 eth1
route add -net 2.0.0.0 netmask 255.0.0.0 gw 20.20.0.3 eth2
route add -net 50.50.0.0 netmask 255.255.255.128 gw 20.20.0.3 eth2
```

Figure 10: Static routes to the topology

4. Setup a DNS server on node dns_root :



```
Corefile
de {
  file db.de
}
org {
  forward . 2.2.2.2
}
.53 {
  file db.root
}

db.de
$ORIGIN de.
$TTL 3600 IN SOA ROOT-SERVER. root.ROOT-SERVER. (
  2017042745 ; serial
  7200 ; refresh (2 hours)
  3600 ; retry (1 hour)
  1209600 ; expire (2 weeks)
  3600 ; minimum (1 hour)
)
$INCLUDE IN A 40.40.0.100

db.root
$ORIGIN .
$TTL 3600 IN SOA ROOT-SERVER. root.ROOT-SERVER. (
  2017042745 ; serial
  7200 ; refresh (2 hours)
  3600 ; retry (1 hour)
  1209600 ; expire (2 weeks)
  3600 ; minimum (1 hour)
)
IN NS ROOT-SERVER.
IN A 1.1.1.1
```

Figure 11: Screenshot of dns setup

5. Add a nameserver And curl gik.de :

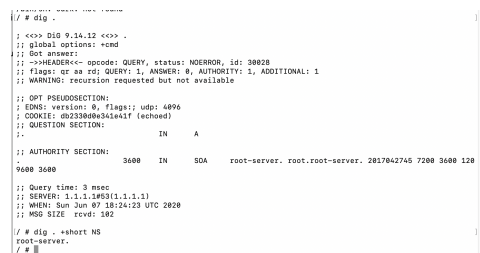


```
moktahid — cd "/Users/moktahid/Desktop/assignment/lab 2/assignment 3/new"...
--- Startup Commands Log
+ ifconfig eth0 10.10.0.10 netmask 255.255.255.0 broadcast 10.10.0.255 up
+ route add default gw 10.10.0.1 eth0
--- End Startup Commands Log

// # curl gik.de
I'm webl
// #
nameserver 1.1.1.1
```

Figure 12: Screenshot of resolv.conf and curl gik.de

6. Use dig to determine the authoritative server for the root domain :



```
// # dig .
; <<>> DIG 9.14.12 c<>> .
;; global options: +cmd
;; Got answer:
;;->HEADER<- opcode: QUERY, status: NOERROR, id: 20820
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 4096
; COOKIE: db23800e341e41f (echoed)
;; QUESTION SECTION:
.                IN      A

;; AUTHORITY SECTION:
.                3600    IN      SOA      root-server. root.root-server. 2017042745 7200 3600 120
          9600 3600

;; Query time: 3 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sun 07 18:24:12 UTC 2020
;; MSG SIZE rcvd: 182

// # dig . -short NS
root-server.
// #
```

Figure 13: Screenshot of authoritative server

7. On node dns_lb setup a DNS servers :

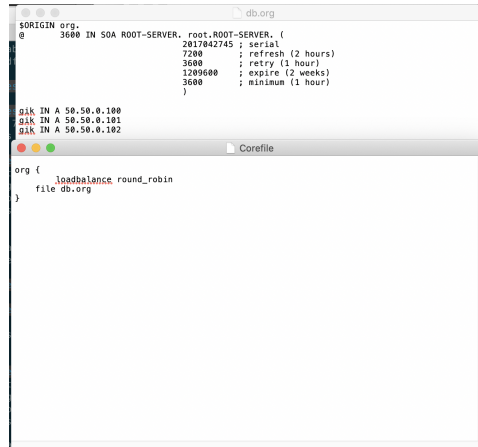


Figure 14: Screenshot of dns_lb setup

8. Modify dns_root to forward the name resolution of org to dns_lb :

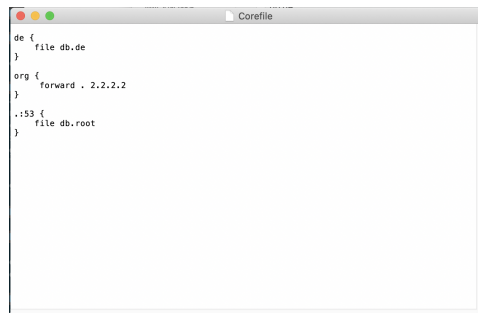


Figure 15: Screenshot of modified dns_root setup

9. Configure CoreDNS on `dns_lb` to load balance the entry `gik.org` :

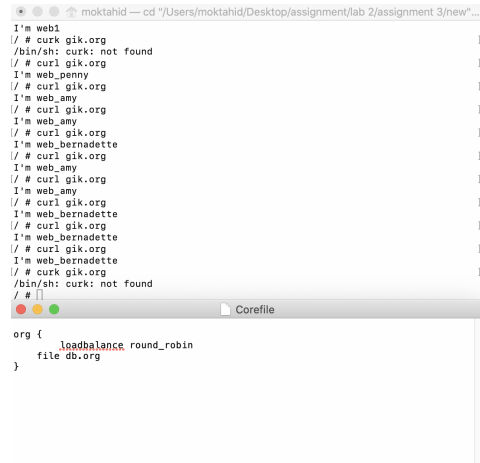


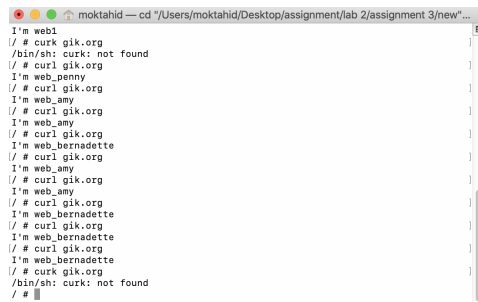
Figure 16: Screenshot of configured core dns and load balancing

Pitfalls of DNS load balancing:

Although easy to implement, round-robin DNS has a number of drawbacks, such as those arising from record caching in the DNS hierarchy itself, as well as client-side address caching and reuse, the combination of which can be difficult to manage. Round-robin DNS should not solely be relied upon for service availability. If a service at one of the addresses in the list fails, the DNS will continue to hand out that address and clients will still attempt to reach the inoperable service.

Round-robin DNS may not be the best choice for load balancing on its own, since it merely alternates the order of the address records each time a name server is queried. Because it does not take transaction time, server load, and network congestion into consideration, it works best for services with a large number of uniformly distributed connections to servers of equivalent capacity. Otherwise, it just does load distribution.[6]

10. dig gik.org :

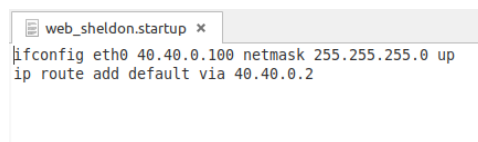


```
moktahid — cd "/Users/moktahid/Desktop/assignment/lab 2/assignment 3/new"...
I'm web1
// # curl gik.org
/bin/sh: curl: not found
// # curl gik.org
I'm web_penny
// # curl gik.org
I'm web_amy
// # curl gik.org
I'm web_amy
// # curl gik.org
I'm web_bernadette
// # curl gik.org
I'm web_amy
// # curl gik.org
I'm web_amy
// # curl gik.org
I'm web_bernadette
// # curl gik.org
I'm web_bernadette
// # curl gik.org
I'm web_bernadette
// # curl gik.org
I'm web_bernadette
// # curl gik.org
/bin/sh: curl: not found
// #
```

Figure 17: Screenshot of dig on gik.org

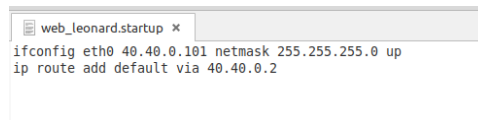
3 Load Balancing with Traefik

1. Followings are the screenshot of new startup files:



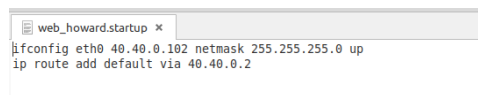
```
web_sheldon.startup x
ifconfig eth0 40.40.0.100 netmask 255.255.255.0 up
ip route add default via 40.40.0.2
```

Figure 18: websheldon.startup file



```
web_leonard.startup x
ifconfig eth0 40.40.0.101 netmask 255.255.255.0 up
ip route add default via 40.40.0.2
```

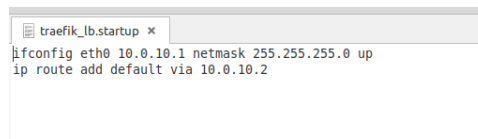
Figure 19: webleonard.startup file



```
web_howard.startup x
ifconfig eth0 40.40.0.102 netmask 255.255.255.0 up
ip route add default via 40.40.0.2
```

Figure 20: webhoward.startup file

2. Startup file for traefik_lb file:



```
traefik_lb.startup x
ifconfig eth0 10.0.10.1 netmask 255.255.255.0 up
ip route add default via 10.0.10.2
```

Figure 21: trafiklbStartup.startup file

listen on port 80 using a file provider:

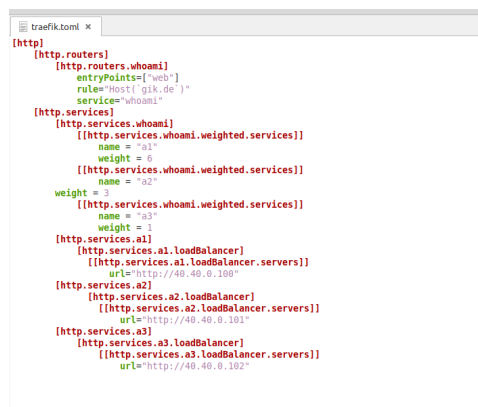


```
traefik.toml x
[entryPoints]
  [entryPoints.web]
    address = ":80"

[providers]
  [providers.file]
    directory = "/conf"
```

Figure 22: 80 Listen Port

Using a file provider forwards requests on gik.de to the new webserver:



```
traefik.toml x
[http]
  [http.routers]
    [http.routers.whoami]
      entryPoints=["web"]
      rule="Host(`gik.de`)"
      service="whoami"
  [http.services]
    [http.services.whoami]
      [[http.services.whoami.weighted.services]]
        name = "a1"
        weight = 5
      [[http.services.whoami.weighted.services]]
        name = "a2"
        weight = 5
      [[http.services.whoami.weighted.services]]
        name = "a3"
        weight = 3
    [http.services.a1]
      [http.services.a1.loadBalancer]
        [[http.services.a1.loadBalancer.servers]]
          url="http://40.40.0.100"
    [http.services.a2]
      [http.services.a2.loadBalancer]
        [[http.services.a2.loadBalancer.servers]]
          url="http://40.40.0.101"
    [http.services.a3]
      [http.services.a3.loadBalancer]
        [[http.services.a3.loadBalancer.servers]]
          url="http://40.40.0.102"
```

Figure 23: File Provider for new webserver

3. Adjust A record of gik de from dns root to point traefik lb

```

dns_root> zones> E dns.de
1 $TTL 60000
2 @ IN SOA dnsde.de. admin.dnsde.de. (2006031201 28800 14400 3600000 0)
3 @ IN NS dnsde.de.
4 dnsde.de. IN A 193.3.1
5
6 gik.de. IN A 10.0.10.0

```

Figure 24: update dns root A record

4. confirm configuration with dig

```

tuhingtuhin ~/resources/classes/XTD/assignment3/lab_config_n_startip dig gik.de
; <<>> DIG 9.11.3-ubuntu1.11-Ubuntu <<>> gik.de
;; global options: +cmd
;; Got answer:
;;->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6450
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;gik.de.                IN      A
;; ANSWER SECTION:
gik.de.                 3600    IN      A      217.160.122.41
;; Query time: 47 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Jun 07 14:41:07 CEST 2020
;; MSG SIZE rcvd: 51
tuhingtuhin ~/resources/classes/XTD/assignment3/lab_config_n_startip

```

Figure 25: dig confirmation

5. Update static route topology

Next we updated our router to suport load balancer traefik for example:

```

E r1.startup
1 ifconfig eth0 10.10.0.1 netmask 255.255.255.0 up
2 ifconfig eth1 30.30.0.1 netmask 255.255.0.0 up
3 ifconfig eth2 20.20.0.1 netmask 255.255.240.0 up
4 ifconfig eth3 1.0.0.1 netmask 255.0.0.0 up
5
6 ip route add 40.40.0.0/24 via 30.30.0.2
7 ip route add 50.50.0.0/25 via 20.20.0.3
8 ip route add 2.0.0.0/8 via 20.20.0.3
9 ip route add 10.0.10.0/24 via 30.30.0.2

```

Figure 26: Update router r1 to adjust traefik

6. Test the load balancing behavior and add a weighted round robin procedure to forward 60 percent, 30 percent and 10 percent request to web sheldon, web leonard and web howard respectively.

```

traefik_lb > conf > @ traefik.toml
1 [http]
2   [http.routers]
3     [http.routers.whoami]
4       entrypoints=["web"]
5       rule="Host(`gik.de`)"
6       service="whoami"
7   [http.services]
8     [http.services.whoami]
9       [[http.services.whoami.weighted.services]]
10        name = "a1"
11        weight = 6
12      [[http.services.whoami.weighted.services]]
13        name = "a2"
14        weight = 3
15      [[http.services.whoami.weighted.services]]
16        name = "a3"
17        weight = 1
18     [http.services.a1]
19       [http.services.a1.loadBalancer]
20       [[http.services.a1.loadBalancer.servers]]
21         url="http://40.40.0.100"
22     [http.services.a2]
23       [http.services.a2.loadBalancer]
24       [[http.services.a2.loadBalancer.servers]]
25         url="http://40.40.0.101"
26     [http.services.a3]
27       [http.services.a3.loadBalancer]
28       [[http.services.a3.loadBalancer.servers]]
29         url="http://40.40.0.102"
30

```

Figure 27: Weighted round robin load balancer configuration

```

/ # curl gik.de
I'm web_leonard
/ # curl gik.de
I'm web_sheldon
/ # curl gik.de
I'm web_sheldon
/ # curl gik.de
I'm web_leonard
/ # curl gik.de
I'm web_howard
/ # curl gik.de
I'm web_sheldon
/ # curl gik.de
I'm web_sheldon
/ # curl gik.de
I'm web_leonard
/ # curl gik.de
I'm web_sheldon
/ # curl gik.de
I'm web_leonard
/ #

```

Figure 28: Test result of load balancer

literature/bib