

Nginx can be used as web server or reverse proxy or load balancer

3 machines:

1. Client1 (Host Only-192.168.100.20) ---> To Host Site
  2. Client2 (Host Only-192.168.100.30) ---> To Host Site
  3. Kali Linux --> Reverse Proxy(Wan-NAT, LAN(Connecting to Backend Servers)-Host Only)
- All Host only are in same LAN Network

*Host Diff Websites on Client1 and Client2 and their ports on 80*

On Client2(192.168.100.30):

```
sudo vi /etc/httpd/conf/httpd.conf
```

Change listen port to 80

```
cd /var/www/html
```

```
vi Index.html
```

Welcome to [www.amazon.com/product](http://www.amazon.com/product)

```
systemctl restart httpd
```

```
curl localhost
```

This should our web page

Now in iptables rule http is allowed from inside to outside as new is not mentioned in rule  
NEW: means allows from outside network to inside network  
and by default for http it's not allowed with NEW

```
[root@localhost html]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination              state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere              anywhere
ACCEPT     icmp --  anywhere              anywhere
ACCEPT     all  --  anywhere              anywhere
ACCEPT     tcp  --  anywhere              anywhere                 state NEW tcp dpt:ssh
REJECT     all  --  anywhere              anywhere                 reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination              reject-with icmp-host-prohibited
REJECT     all  --  anywhere              anywhere
```

As shown NEW is used for SSH as we access SSH from outside

So Now we have to access websites from outside so add NEW to http i.e TCP 80 rule

So Add Rule

```
iptables -I INPUT 2 -p tcp --dport 80 -m state --state NEW,ESTABLISHED,RELATED -j
ACCEPT
```

```
[root@localhost html]# iptables -I INPUT 2 -p tcp --dport 80 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
[root@localhost html]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination              state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere              anywhere                 tcp dpt:http state NEW,RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere              anywhere
ACCEPT     all  --  anywhere              anywhere
ACCEPT     tcp  --  anywhere              anywhere                 state NEW tcp dpt:ssh
REJECT     all  --  anywhere              anywhere                 reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination              reject-with icmp-host-prohibited
REJECT     all  --  anywhere              anywhere
```

On Client1(192.168.100.20): Same as Client2

```

kali linux x CentOs 9 Client 1 x CentOs 9 client 2 x
Activities Terminal Nov 12 18:10
root@localhost:/var/www/html

[root@localhost ~]# sudo vi /etc/httpd/conf/httpd.conf
[root@localhost ~]# cd /var/www/html
[root@localhost html]# vi index.html
[root@localhost html]# systemctl restart httpd
[root@localhost html]# curl localhost
Welcome to Client1!

[root@localhost html]#

```

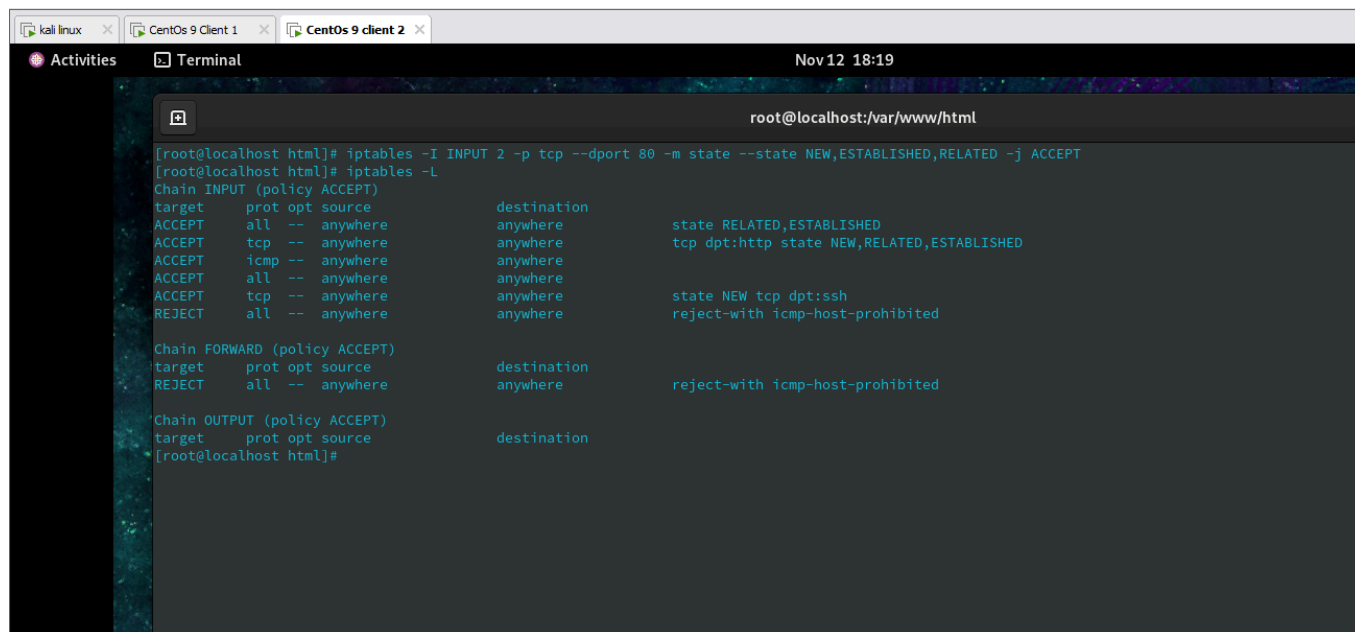
in index.html enter:

Welcome to [www.amazon.com](http://www.amazon.com)

```
iptables -I INPUT 2 -p tcp --dport 80 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

To check if rule is added:

```
iptables -L
```



```
root@localhost:/var/www/html
[root@localhost html]# iptables -I INPUT 2 -p tcp --dport 80 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
[root@localhost html]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination           state RELATED,ESTABLISHED
ACCEPT    tcp  --  anywhere              anywhere              tcp dpt:http state NEW,RELATED,ESTABLISHED
ACCEPT    icmp --  anywhere              anywhere
ACCEPT    all  --  anywhere              anywhere
ACCEPT    tcp  --  anywhere              anywhere              state NEW tcp dpt:ssh
REJECT    all  --  anywhere              anywhere              reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination           reject-with icmp-host-prohibited
REJECT    all  --  anywhere              anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@localhost html]#
```

Now try to access each other website like:

From Client1:

```
curl 192.168.100.30
```

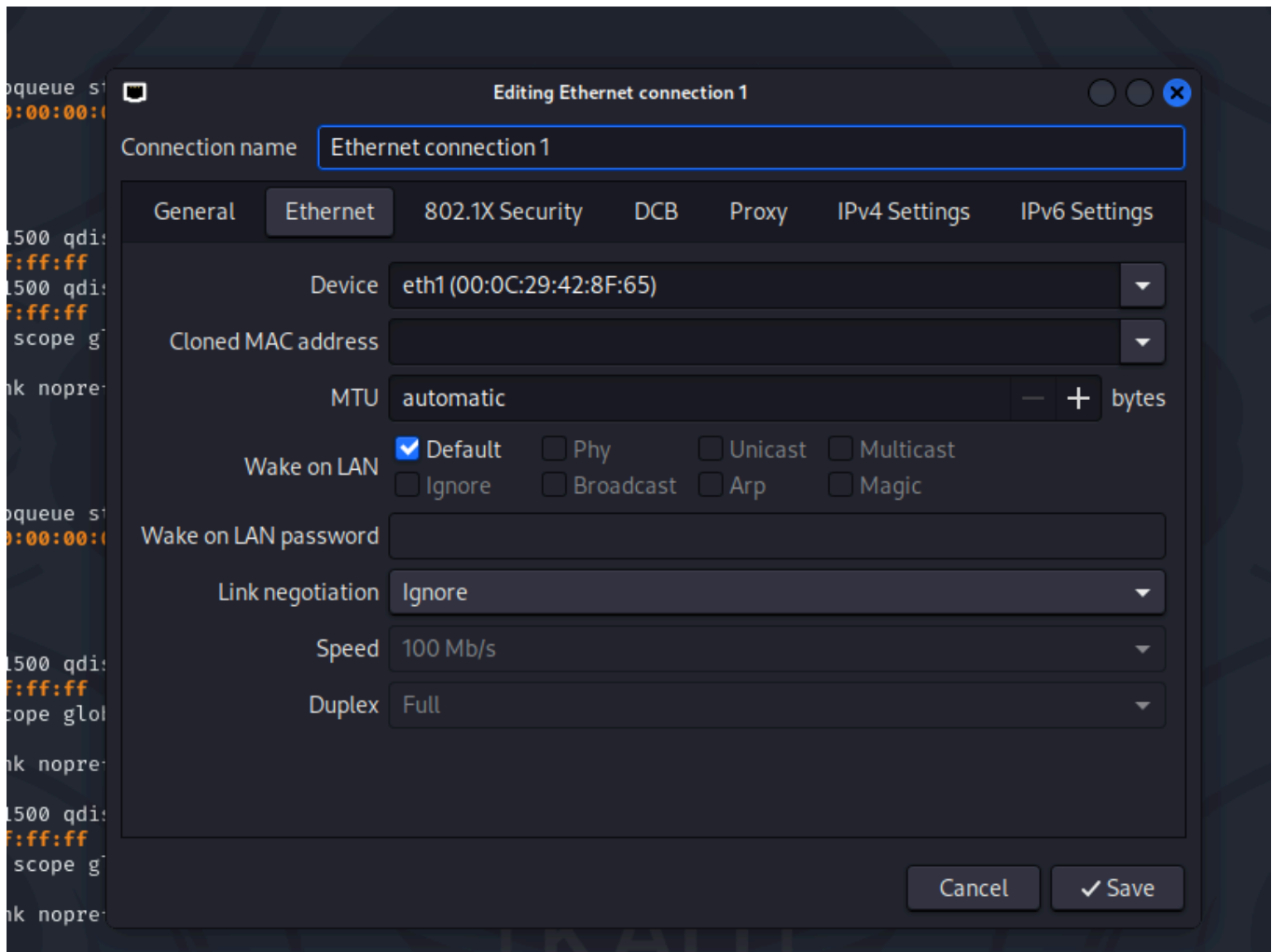
From Client2:

```
curl 192.168.100.20
```

On Kali:

Setting --> Advanced Network Config -->

1. Select wiredConnection1 (For WAN(NAT)) --> Select Adapter eth0 --> save
2. Click on add at bottom --> Select Adapter eth1 --> save



ip a

Host only(eth1) ----> same Network as of our Clients (192.168.100.128)  
 NAT(eth0) ----> Wan (192.168.75.139)

```
(ditiss@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 00:0c:29:42:8f:5b brd ff:ff:ff:ff:ff:ff
   inet 192.168.75.139/24 brd 192.168.75.255 scope global dynamic noprefixroute eth0
       valid_lft 1415sec preferred_lft 1415sec
   inet6 fe80::da0d:965a:173a:6d3d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 00:0c:29:42:8f:65 brd ff:ff:ff:ff:ff:ff
   inet 192.168.100.128/24 brd 192.168.100.255 scope global dynamic noprefixroute eth1
       valid_lft 1360sec preferred_lft 1360sec
   inet6 fe80::69d7:49b8:4c16:b19c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Now Install Nginx:

```
sudo apt install nginx -y
```

If this not work:

```
Sudo apt update -y
```

```
cd /etc/nginx
```

We need to create a copy of original file so if anything goes wrong we can revert back

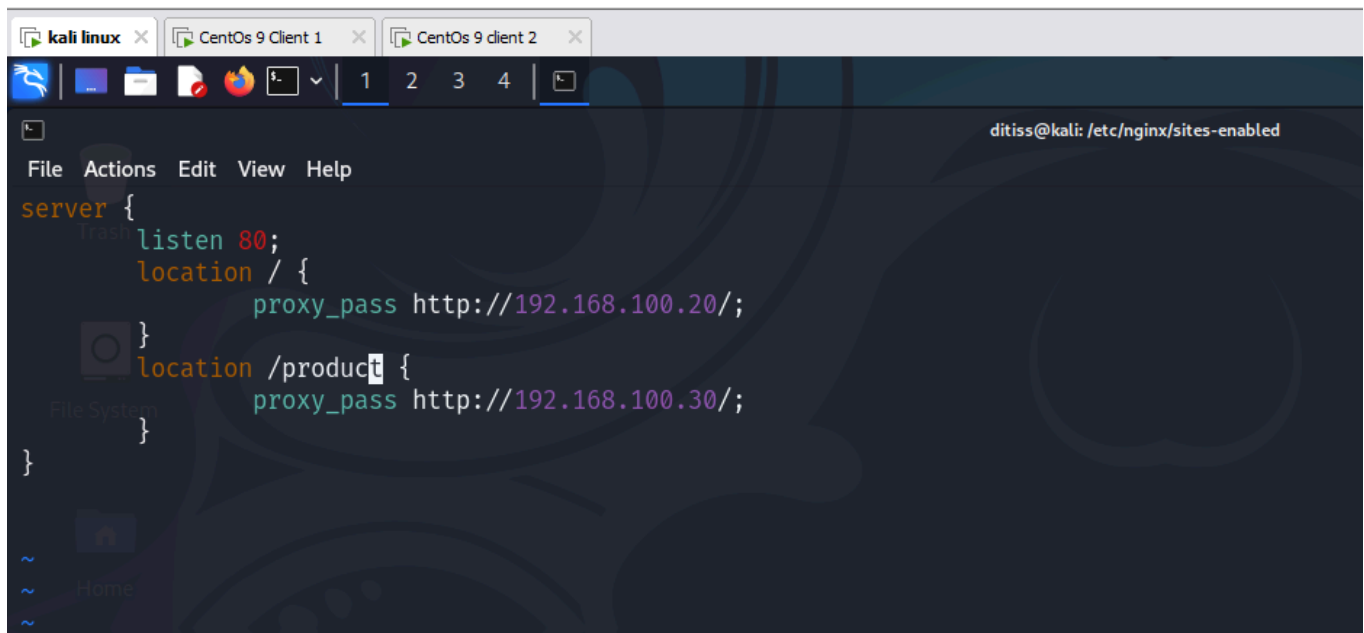
```
sudo cp nginx.conf nginx.conf.original
```

```
sudo vi /etc/nginx/nginx.conf
```

```
cd sites-enabled
```

```
sudo vi rproxy.conf
```

Enter in this file below data:



```
server {  
    listen 80;  
    location / {  
        proxy_pass http://192.168.100.20/;  
    }  
    location /product {  
        proxy_pass http://192.168.100.30/;  
    }  
}
```

```
http{
```

```
}
```

can be only defined once and that is already define in nginx.conf file

so here only server is used

Now if try curl 192.168.75.139

it shows default page but we want of client1

```
(ditiss@kali)-[/etc/nginx/sites-enabled]
$ ls
default  rproxy.conf

(ditiss@kali)-[/etc/nginx/sites-enabled]
$ ls -all
total 12
drwxr-xr-x 2 root root 4096 Nov 12 20:34 .
drwxr-xr-x 8 root root 4096 Nov 12 20:23 ..
lrwxrwxrwx 1 root root   34 Nov 11 17:44 default -> /etc/nginx/sites-available/default
-rw-r--r-- 1 root root  139 Nov 12 20:32 rproxy.conf

(ditiss@kali)-[/etc/nginx/sites-enabled]
$
```

Sites are actually made/configured in sites-available and then by creating symlink they are enabled through sites-enabled

But here we have made our sites on client1 and Client2 but on proxy we just need to enable them

but also the symlink named as "default" in sites-enabled is not allowing to enable our client1 and 2 sites so we will remove default from sites-enabled only

```
sudo rm -f default
```

```
sudo systemctl restart nginx
```

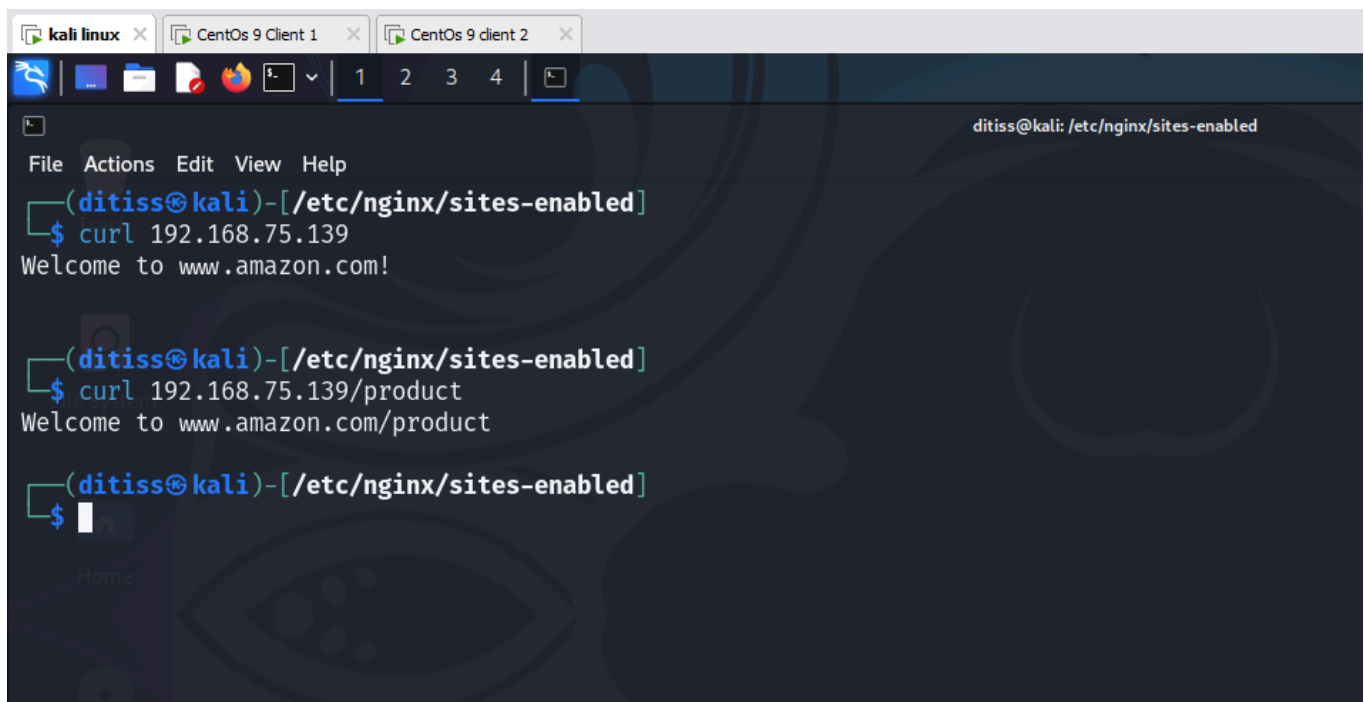
Now through WAN of proxy i.e from outside network we can access our sites

```
curl 192.168.75.139
```

```
Welcome to www.amazon.com
```

```
curl 192.168.75.139/product
```

```
Welcome to www.amazon.com/product
```



The image shows a terminal window with three tabs: 'kali linux', 'CentOs 9 Client 1', and 'CentOs 9 client 2'. The active tab is 'kali linux'. The terminal displays the following commands and outputs:

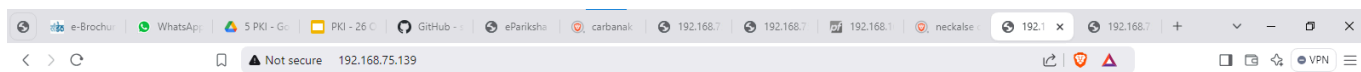
```
ditiss@kali: /etc/nginx/sites-enabled
File Actions Edit View Help
(ditiss@kali)-[/etc/nginx/sites-enabled]
$ curl 192.168.75.139
Welcome to www.amazon.com!

(ditiss@kali)-[/etc/nginx/sites-enabled]
$ curl 192.168.75.139/product
Welcome to www.amazon.com/product

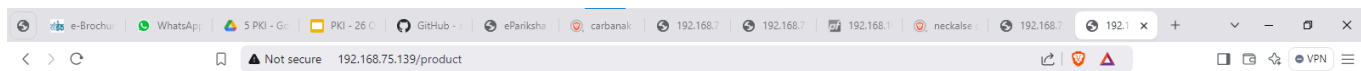
(ditiss@kali)-[/etc/nginx/sites-enabled]
$
```

The terminal background features a Kali Linux logo and the word 'Home'.

Also Check from windows browser



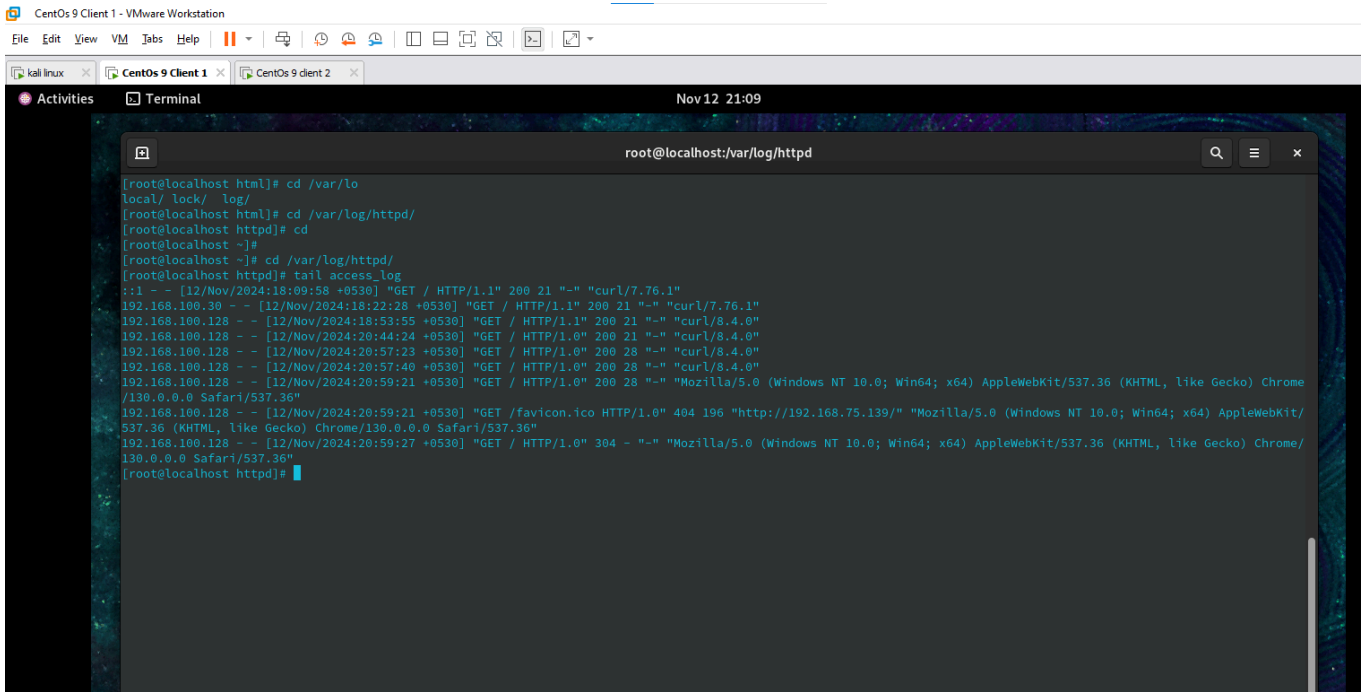
Welcome to www.amazon.com!



Welcome to www.amazon.com product

Now we can check logs of client1(backend server)  
logs contains IP of proxy LAN and not of outside client





So to forward outside clients IP:

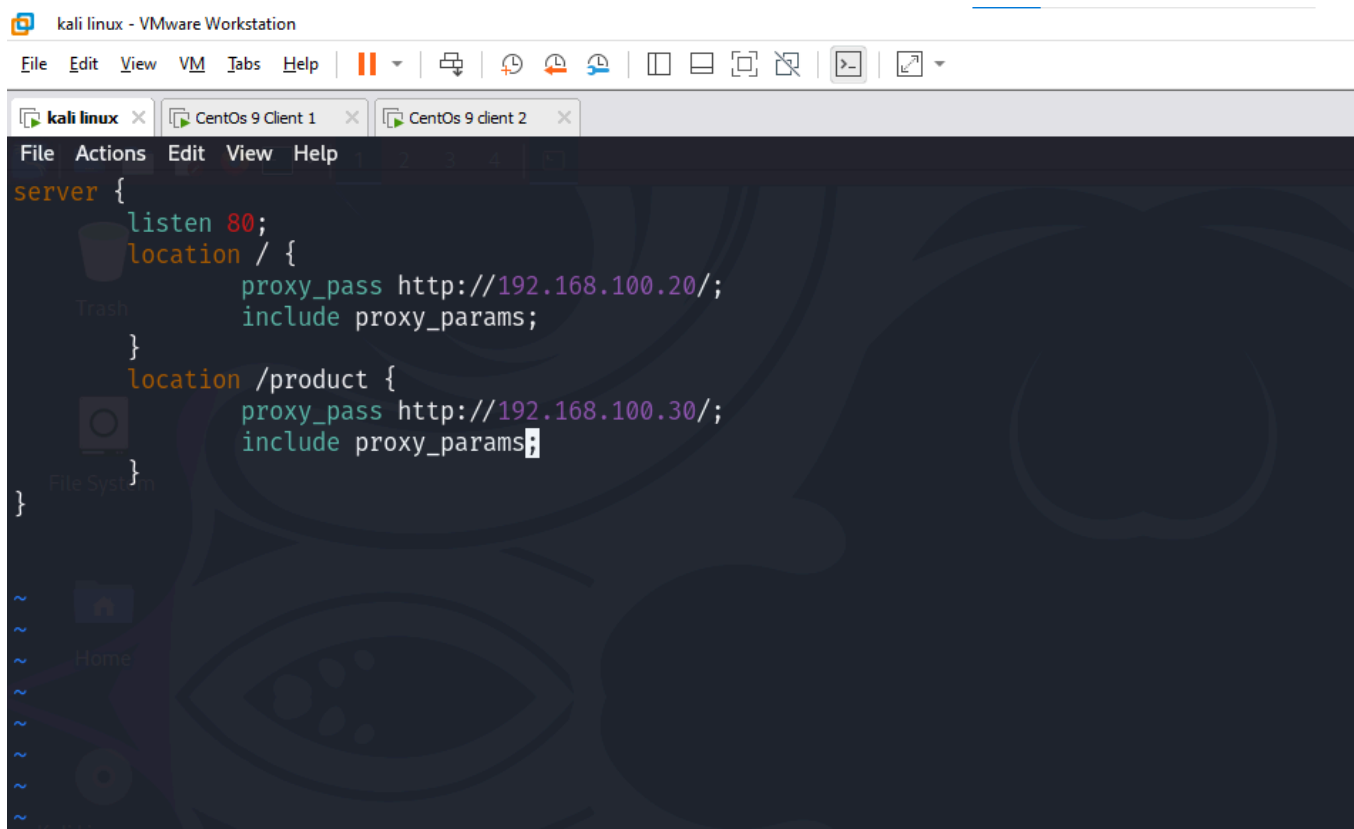
```
cd /etc/nginx/  
cat proxy_params
```

```
proxy_set_header Host $http_host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
```

we need to add the parameters to rproxy file

```
cd /etc/nginx/sites-enabled
```

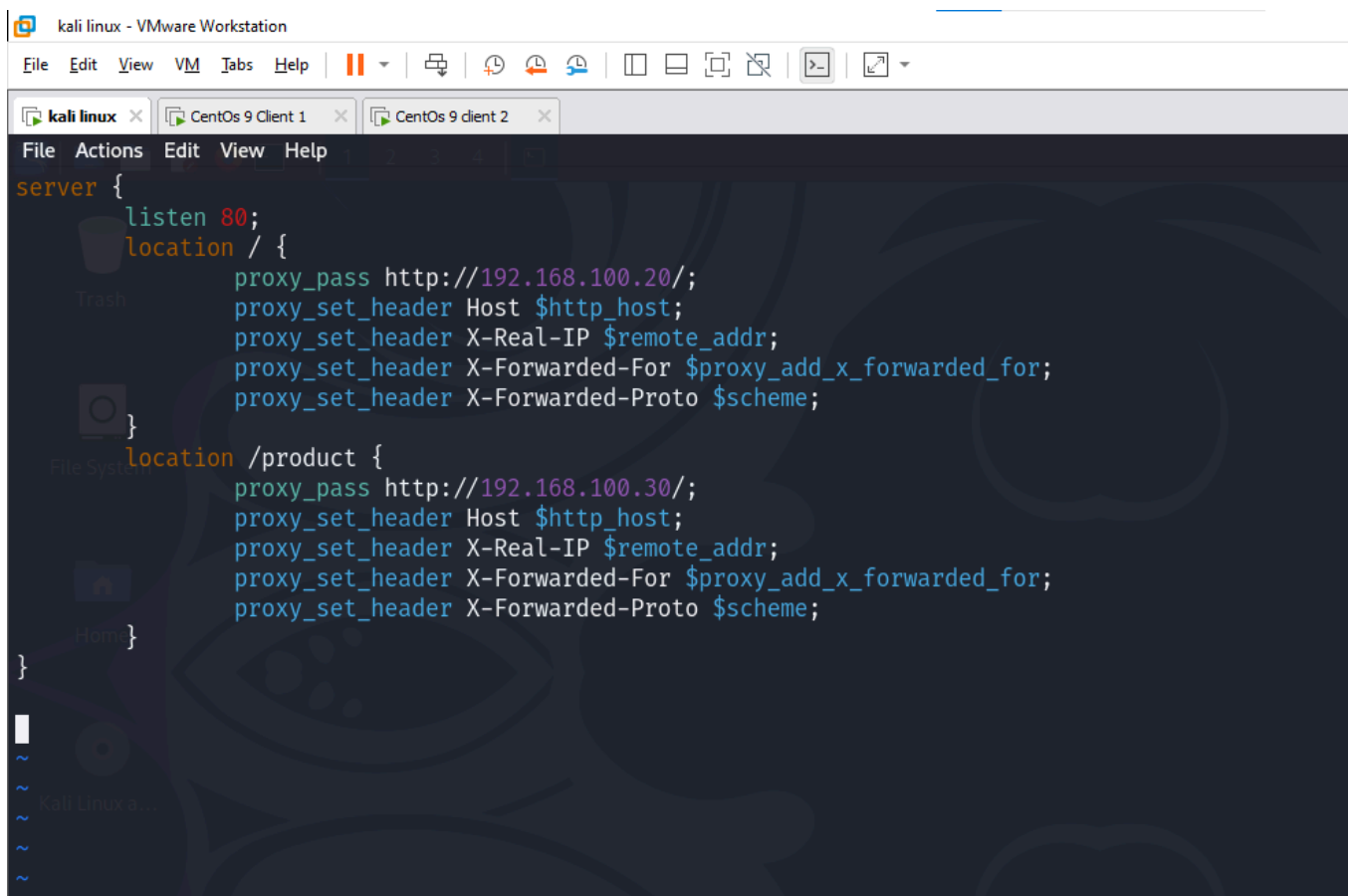
```
sudo vi rproxy.conf
```



The screenshot shows a Kali Linux terminal window within a VMware Workstation. The terminal displays the configuration for an Nginx server. The configuration includes a 'server' block with a 'listen' directive on port 80. There are two 'location' blocks: one for the root path '/' and another for '/product'. Both locations use 'proxy\_pass' to forward requests to different upstream servers and include a 'proxy\_params' module. The background of the terminal window features a Kali Linux logo.

```
server {
    listen 80;
    location / {
        proxy_pass http://192.168.100.20/;
        include proxy_params;
    }
    location /product {
        proxy_pass http://192.168.100.30/;
        include proxy_params;
    }
}
```

OR



The screenshot shows a Kali Linux terminal window within a VMware Workstation, displaying a more detailed Nginx configuration. This version includes 'proxy\_set\_header' directives for each location to preserve client information. The configuration is similar to the first one but adds headers for Host, X-Real-IP, X-Forwarded-For, and X-Forwarded-Proto. The background of the terminal window features a Kali Linux logo.

```
server {
    listen 80;
    location / {
        proxy_pass http://192.168.100.20/;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
    location /product {
        proxy_pass http://192.168.100.30/;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Now go to client1:

```
cd /var/log/httpd/
```

```
tail access_log
```

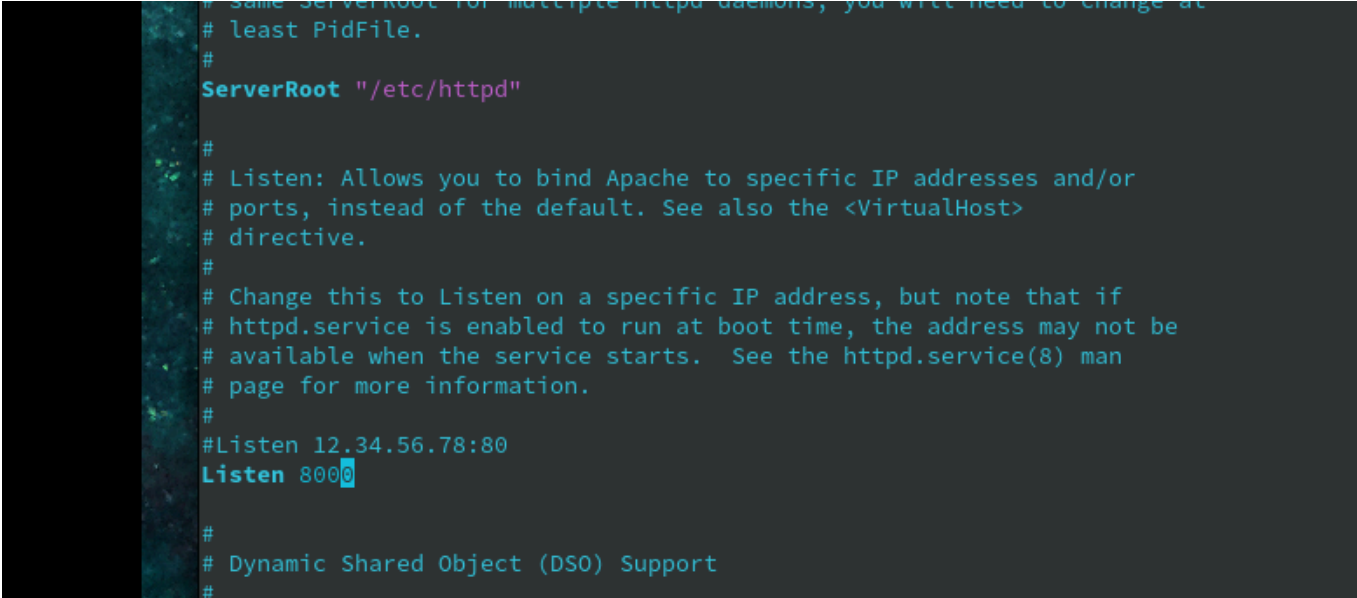
Now also we can't see outside clients IP because we are using VM and packets

```
sudo systemctl restart nginx
```

Now we will Change port of Client1 to 8000

```
sudo vi /etc/httpd/conf/httpd.conf
```

```
listen 8000
```



```
# Same ServerRoot for multiple httpd daemons, you will need to change at  
# least PidFile.  
#  
ServerRoot "/etc/httpd"  
  
#  
# Listen: Allows you to bind Apache to specific IP addresses and/or  
# ports, instead of the default. See also the <VirtualHost>  
# directive.  
#  
# Change this to Listen on a specific IP address, but note that if  
# httpd.service is enabled to run at boot time, the address may not be  
# available when the service starts. See the httpd.service(8) man  
# page for more information.  
#  
#Listen 12.34.56.78:80  
Listen 8000  
  
#  
# Dynamic Shared Object (DSO) Support  
#
```

```
sudo setenforce 0
```

```
systemctl restart httpd
```

```
sudo iptables -I INPUT 2 -p tcp --dport 8000 -j ACCEPT
```

```
curl localhost:8000
```

Now we will Change port of Client2 to 9000

```
sudo vi /etc/httpd/conf/httpd.conf
```

```
Listen 9000
```

```
sudo setenforce 0
```

```
systemctl restart httpd
```

```
sudo iptables -I INPUT 2 -p tcp --dport 9000 -j ACCEPT
```

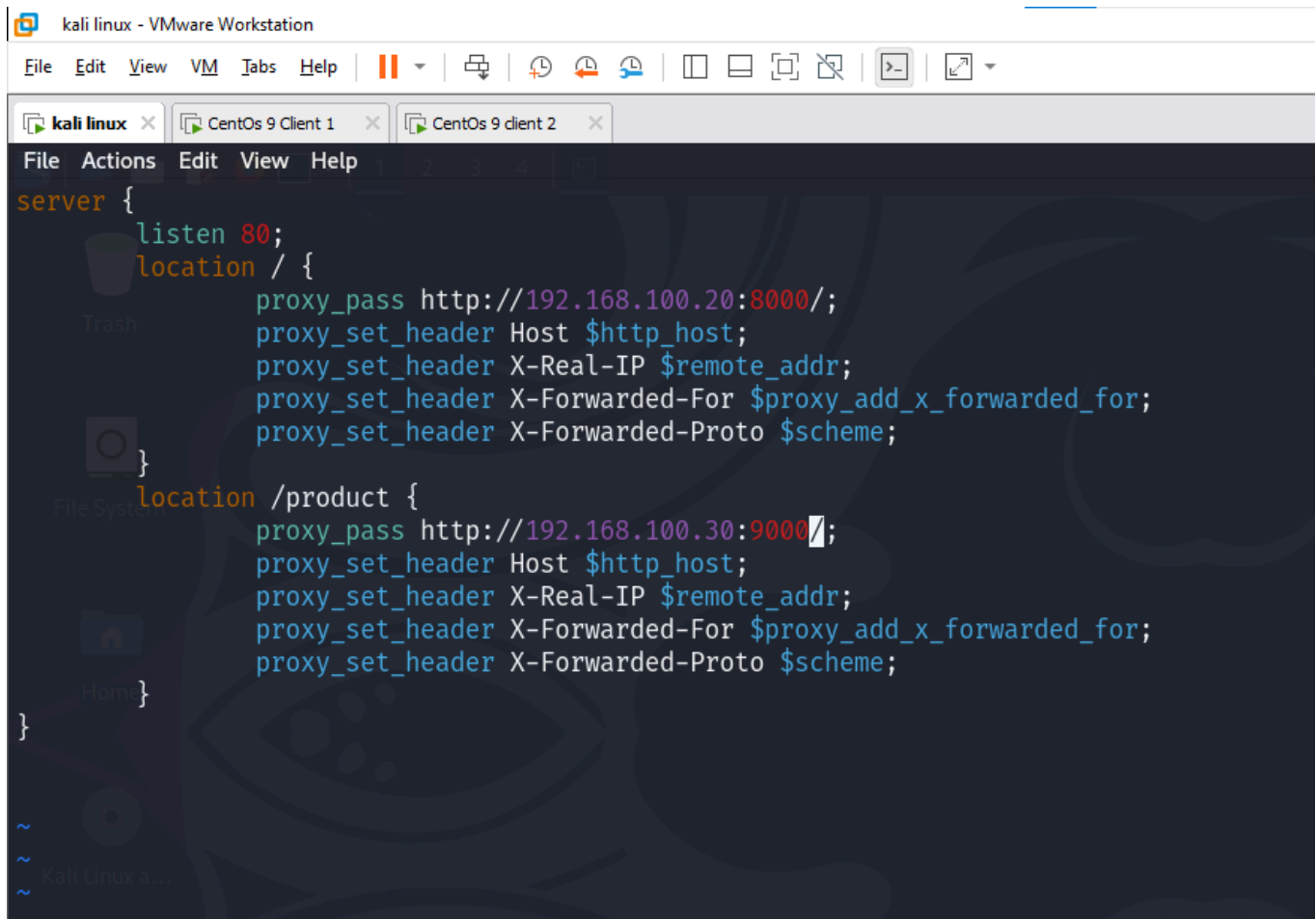
```
curl localhost:9000
```

Now go to Kali:

```
cd /etc/nginx/sites-enabled
```

```
sudo vi rproxy.conf
```

In rproxy file give port no:



```
sudo systemctl restart nginx
```

```
curl 192.168.75.139
```

```
Welcome to www.amazon.com!
```

```
curl 192.168.75.139/product
```

```
Welcome to www.amazon.com/product
```