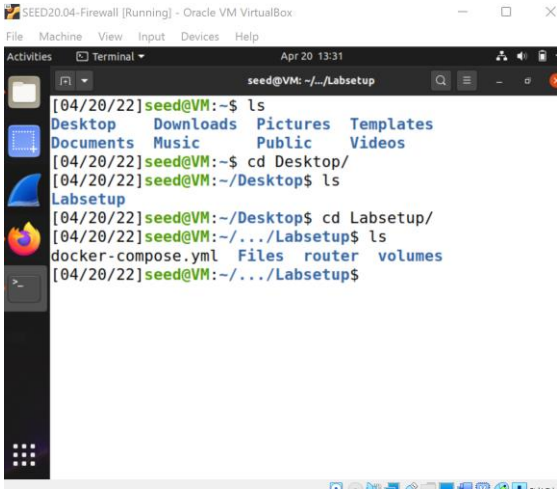
Lab 5 Firewall Exploration

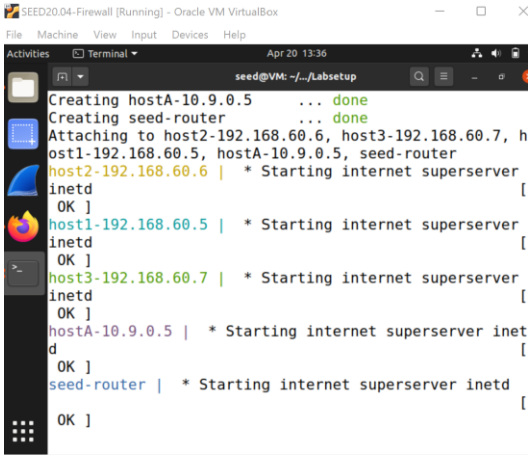
I. Lab set up

a. Prebuild 20.04 Seed image on VirtualBox

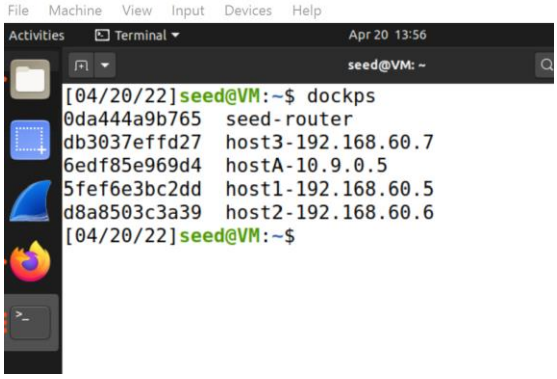
b. Download Labsetup.zip on the VM

c. 

d. Run dcbuild and dcup to build and start the container.

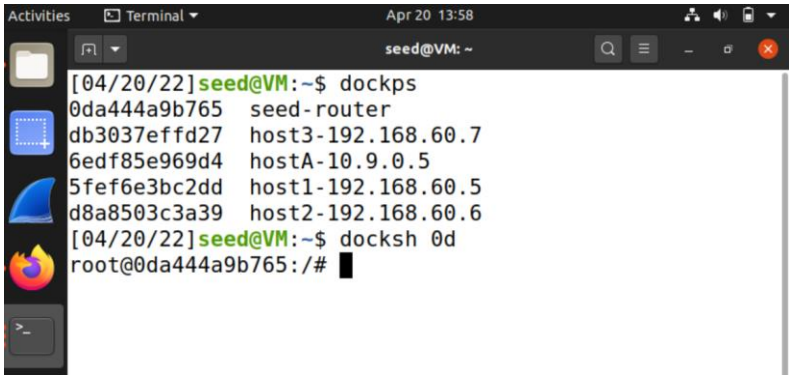
e. 

f. Open another terminal to run dockps to check for IDs.

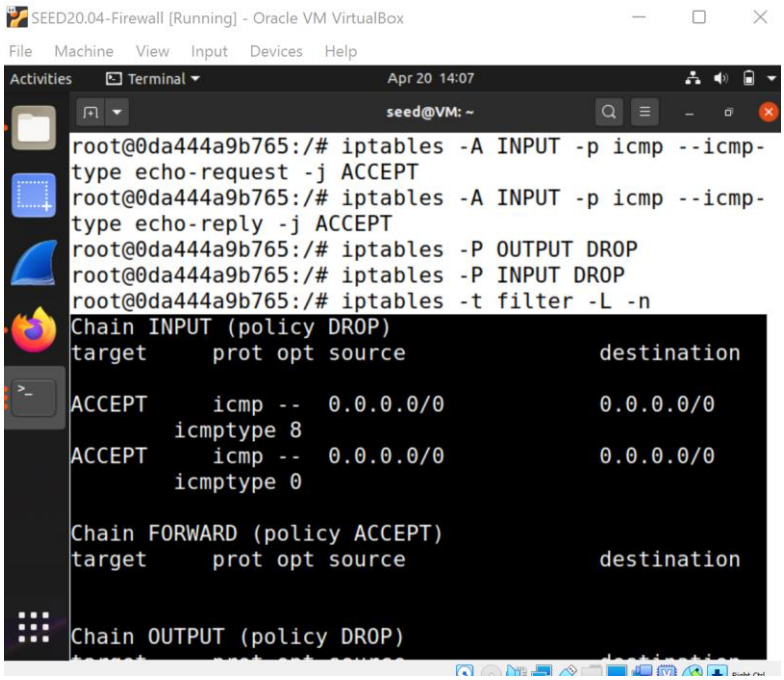
g. 

II. Task 2.A

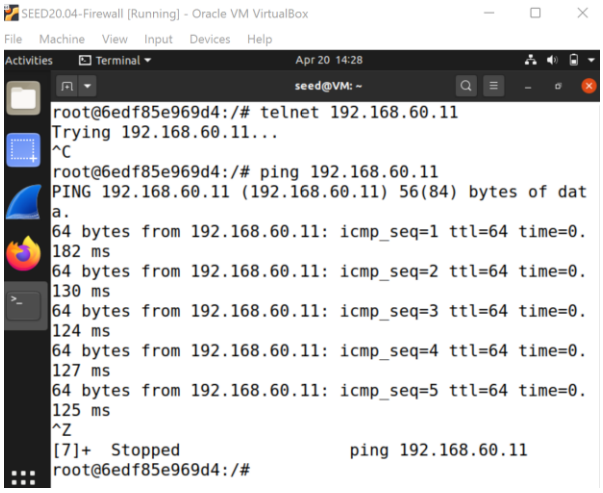
a. Get a shell on seed-router.

b. 

c. Execute the provided commands and check the rules in the table.

d. 

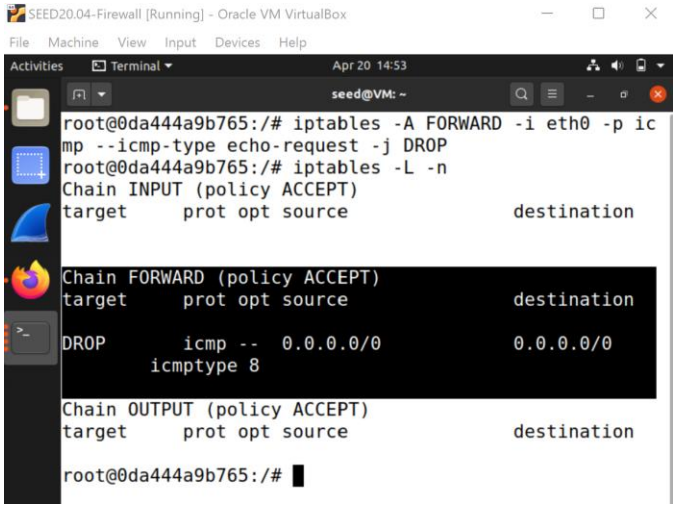
e. On the shell of 10.9.0.5 try telnet and ping the server. Telnet does not work anymore yet ping would still work since the commands filtered all incoming and outcoming traffic except ping (ICMP packets).

f. 

g. Restart the container to restore states of all tables.

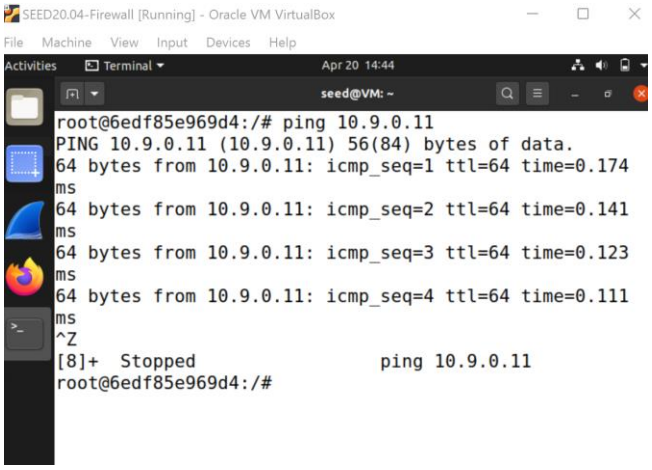
III. Task 2.B

a. Run the command in router's shell and check the table.

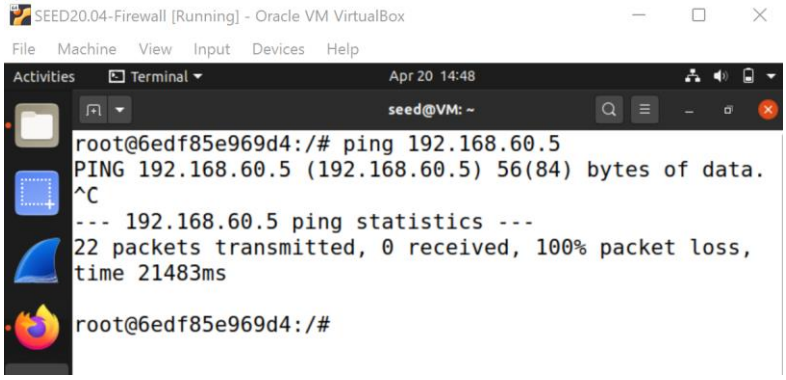
b. 

c. Check

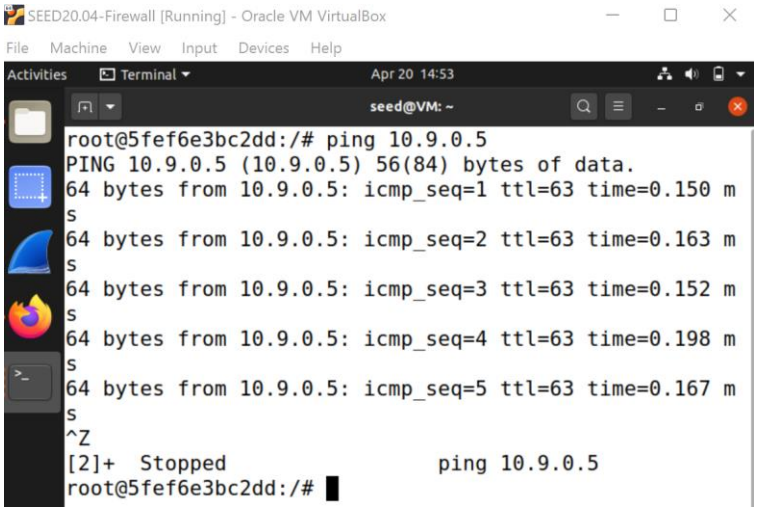
i. Outside host (host A: 10.9.0.5) can ping the router.

ii. 

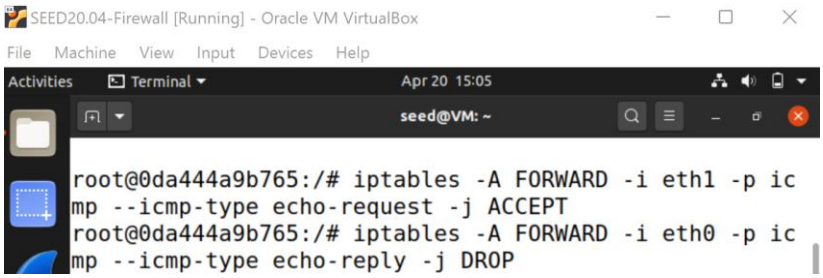
iii. Outside host (host A: 10.9.0.5) cannot ping internal hosts (host 1: 192.168.60.5).

iv. 

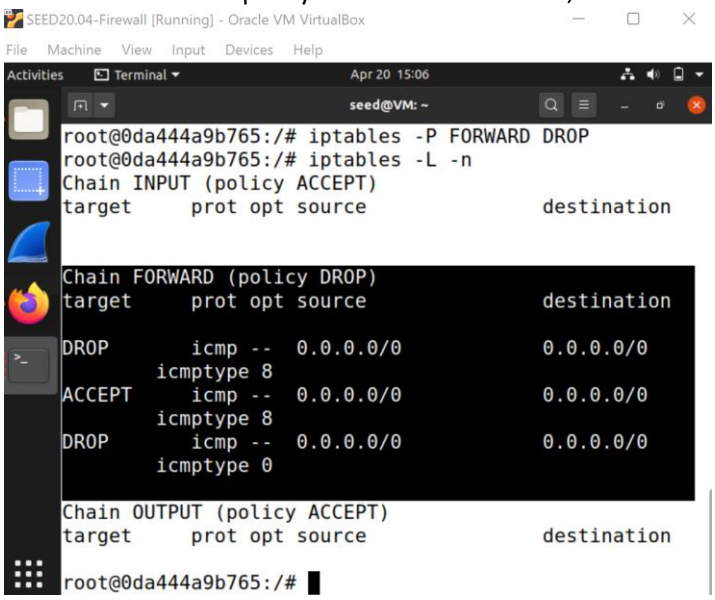
v. Internal host (host 1: 192.168.60.5) can ping outside host (host A: 10.9.0.5).

vi. 

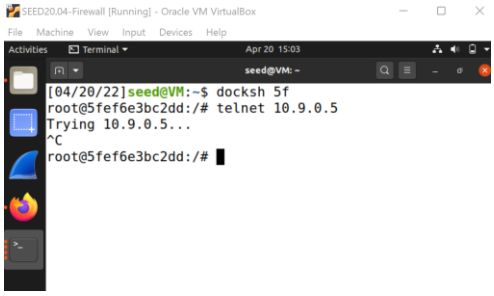
vii. To prevent other packets between internal and external host, added more rules to the table.

viii. 

ix. Also set the default policy for Forward to DROP, and check the table.

x. 

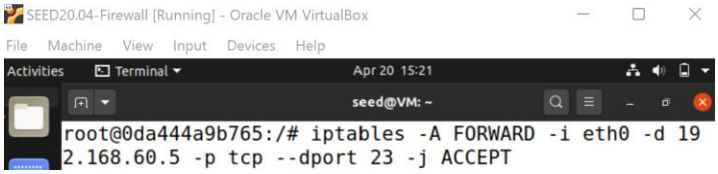
xi. Telnet outside host (host A: 10.9.0.5) from Internal host (host 1: 192.168.60.5). Packets should be dropped, and telnet should not work.

xii. 

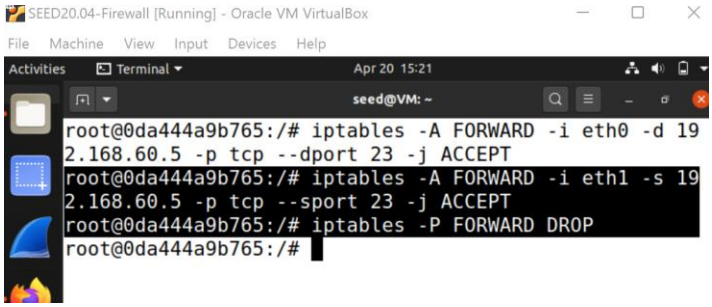
xiii. The firewall works as expected. Restart the container to clean up the rules.

IV. Task 2.c

a. Add the following rule so that all the internal hosts run a telnet server (listening to port 23). Outside hosts can only access the telnet server on 192.168.60.5, not the other internal hosts.

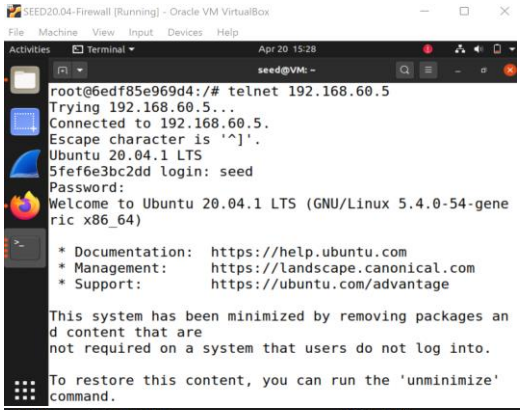
b. 

c. Add the following rule so that internal hosts can access all the internal servers and internal hosts cannot access external servers.

d. 

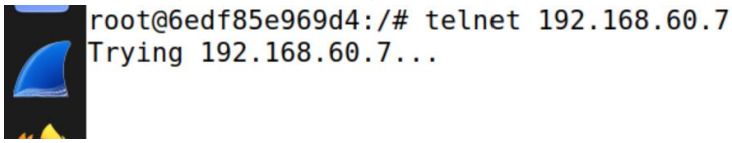
e. Check

i. Outside hosts (host A: 10.9.0.5) can only access the telnet server on 192.168.60.5, not the other internal hosts.

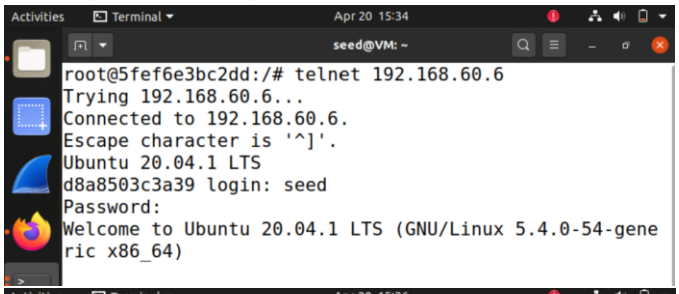
ii. 

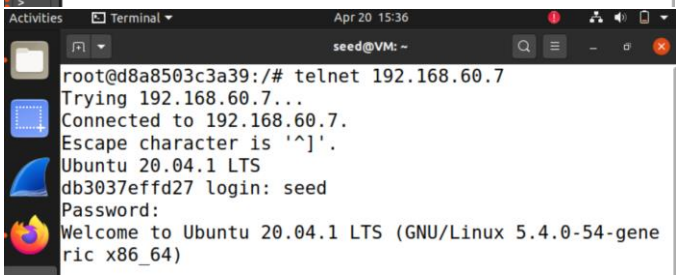
iii. 

iv. Outside hosts (host A: 10.9.0.5) cannot access other internal servers.

v. 

vi. . Internal hosts can access all the internal servers. Here I will check if internal host 1 (192.168.60.5) can access host 2 (192.168.60.6) and host 2 can access host 3 (192.168.60.7).

vii. 

viii. 

ix. Internal hosts cannot access external servers. Here I will check if internal host 1(192.168.60.5) host2(192.168.60.6) and host 3(192.168.60.7) can access host A(10.9.0.5).

x. 

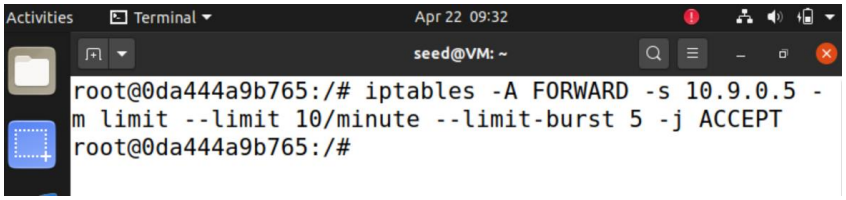
xi. 

xii. 

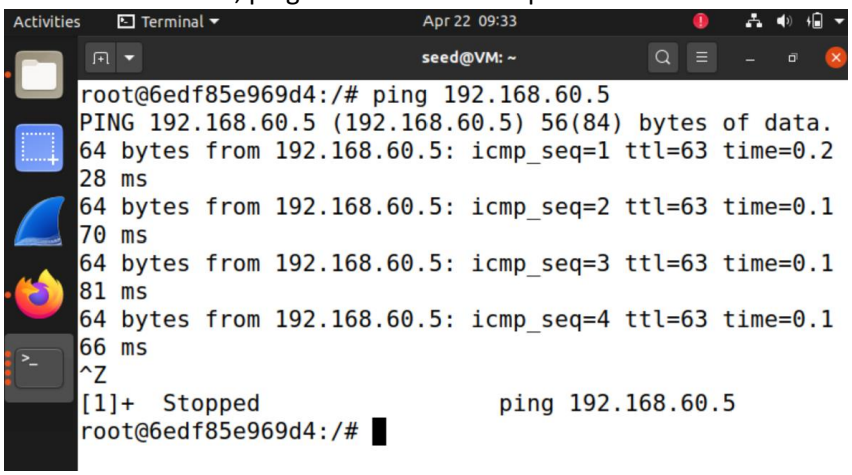
xiii. The rules to protect the TCP server was set up successfully. Restart the docker to clean up the tables.

V. Task 4

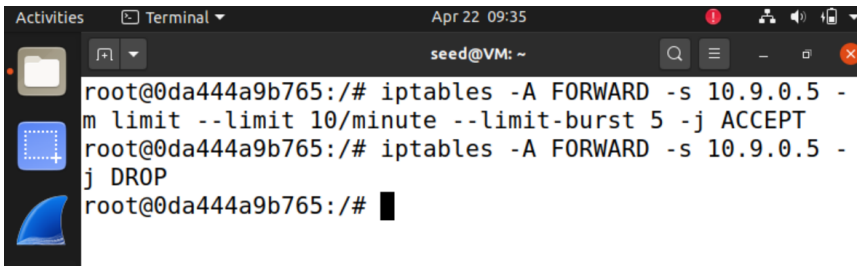
a. In the router shell, add the first rule only.

b. 

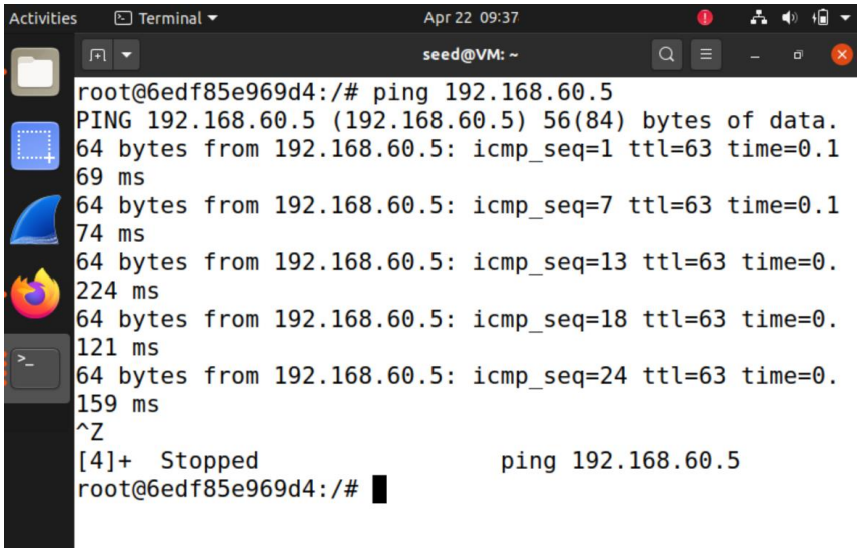
c. On shell of 10.9.0.5, ping 192.168.60.5. The packets were transmitted.

d. 

e. Add the second rule.

f. 

g. Ping again, and with second rule, the limit takes effect as the icmp_seq are not consecutive since the rule limits to 10 packets per minute. The second rule is needed since it specifies the default rule for forward is to drop the packets.

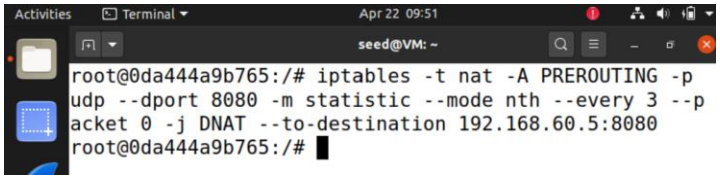
h. 

VI. Task 5

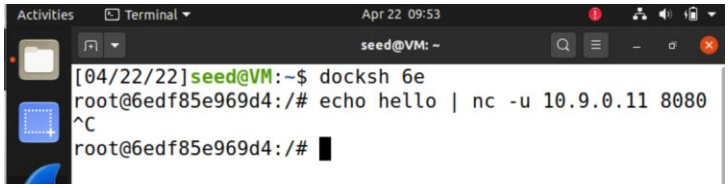
a. Run nc -luk 8080 on host 1-3

b. Using round robin:

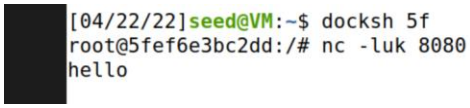
c. On router shell, run following command.

d. 

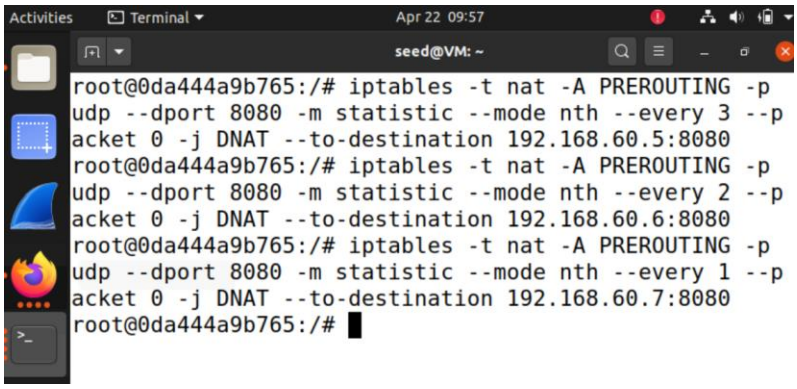
e. On host A shell, run following command.

f. 

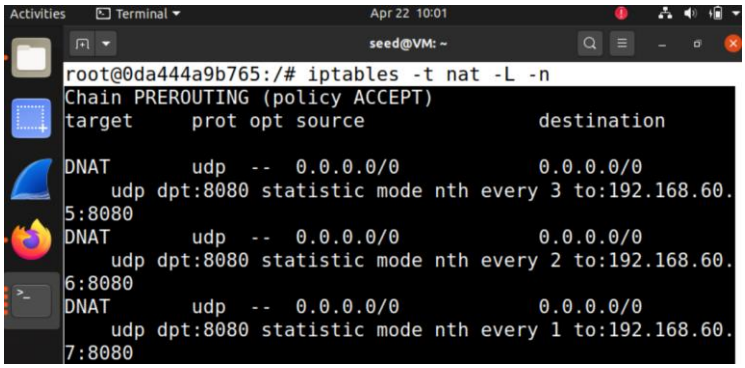
g. On host 1, we can see 'hello' printed out.

h. 

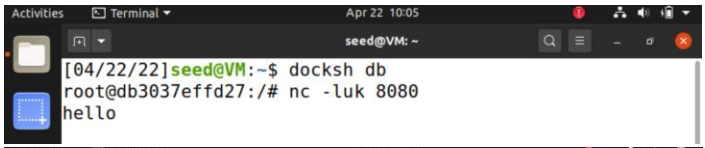
i. In order to all the three internal hosts get the equal number of packets, two more rules are needed. The rule added previously sent the first packet among three packets to host1. We need to dispatch the 2 packets left. The first added rule specifies that every 2 packets, send the first packet to host 2. The last rule specifies that for the third packet, send to host 3.

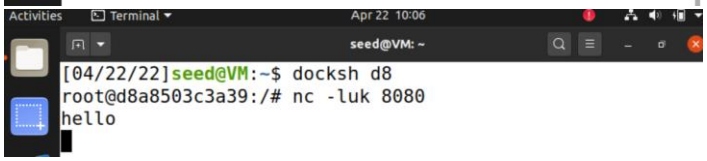
j. 

k. Check the nat table rules:

l. 

m. Now on host A run echo hello three times, and host 1-3 will printout hello.

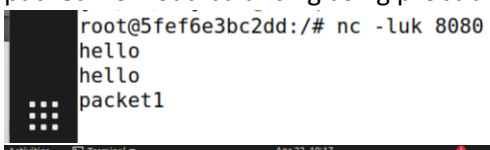
n. 

o. 

p. Using the random node:

q. Restart router and run following command in router shell. For host 1, the probability is 1/3 since it is receiving one packets out of three. For host2, the probability is ½, since it is receiving one packets out of two, and for host 3, the probability is 1.

r. On host A, send three packets: packet 1-3 and host 1-3 will receive correspondingly packet 1-3. Load balancing using probability mode was implemented successfully.

s. 

t. 

u.