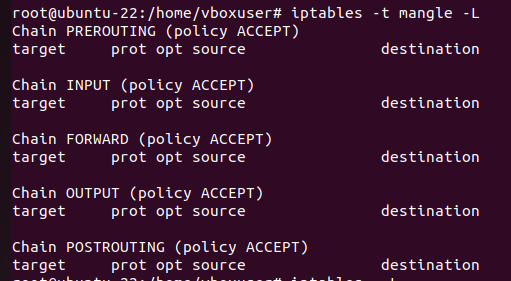
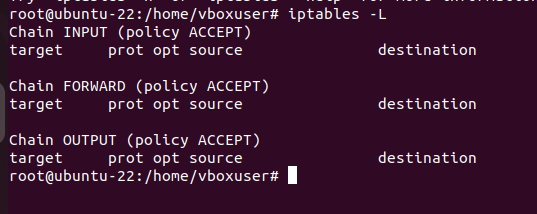
|  |  |
| --- | --- |
| **Name** | **Hatim Sawai** |
| **UID no.** | **2021300108** |
| **Experiment No.** | 8 |
| **AIM** | **Configure Firewall rules using IP tables. Upload the compressed file as per the instruction given in one page manual of this experiment.** |

IP Tables Project Exercises (all questions are with respect to the filter table, and are to be executed independent of the other questions, unless otherwise noted. So, remember to flush the iptables after executing each question, unless you are asked to follow up from a previous question):

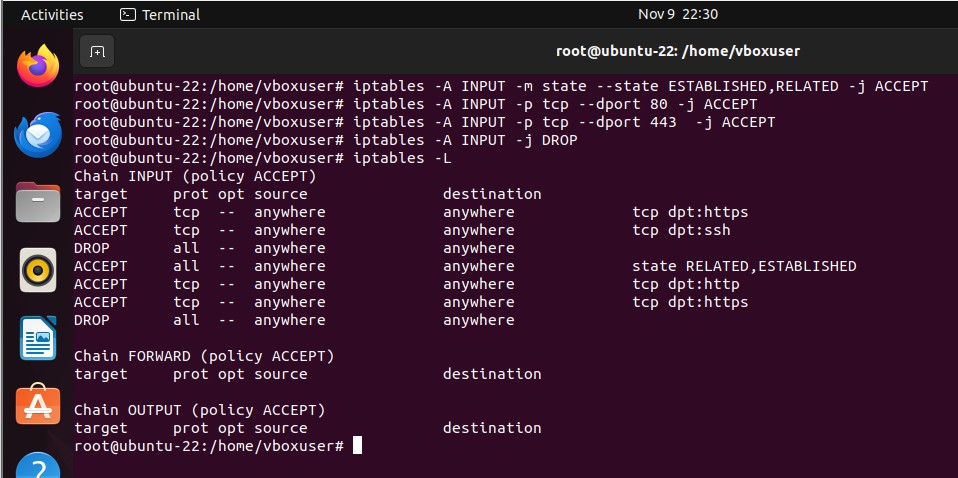
Q1) Set the default policy for the INPUT chain to DROP. The firewall should only allow incoming packets from the network prefix 143.132.0.0/16. The default policy for the OUTPUT chain is ACCEPT. So, the user working on the machine could visit any website like [www.google.com.](http://www.google.com/) Given the above policy for incoming packets, can the web pages visited by the user be displayed in the browser? Explain



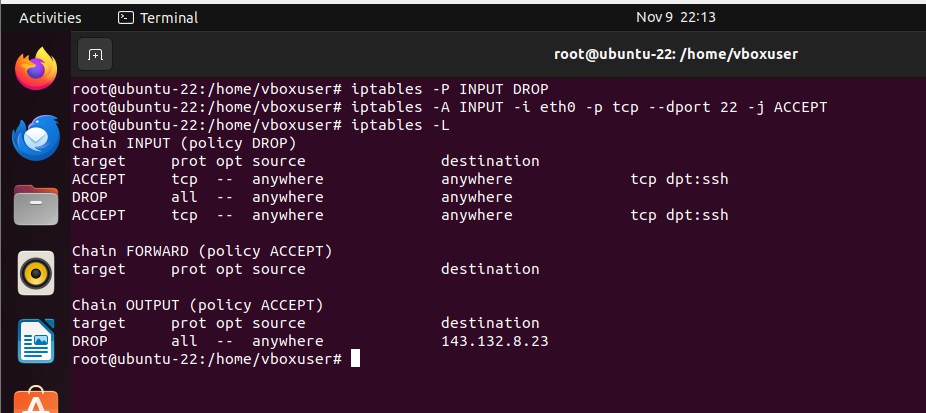


Q2) Set the default policy for the INPUT chain be DROP and the default policy for the OUTPUT chain be ACCEPT. Configure the INPUT chain to accept all incoming web traffic to port 80 and drop any other incoming traffic. Can you visit the website: [www.hotmail.com?](http://www.hotmail.com/) Why or why not? If you cannot visit the website, what aspect of this website is preventing you from visiting it, given that your default OUTPUT policy is ACCEPT and the firewall has been configured to accept traffic coming to port 80? Also, if you cannot visit the website, configure the firewall to let you be able to visit websites of such type. What changes/deletions/additions to the rules had to be done to facilitate this?

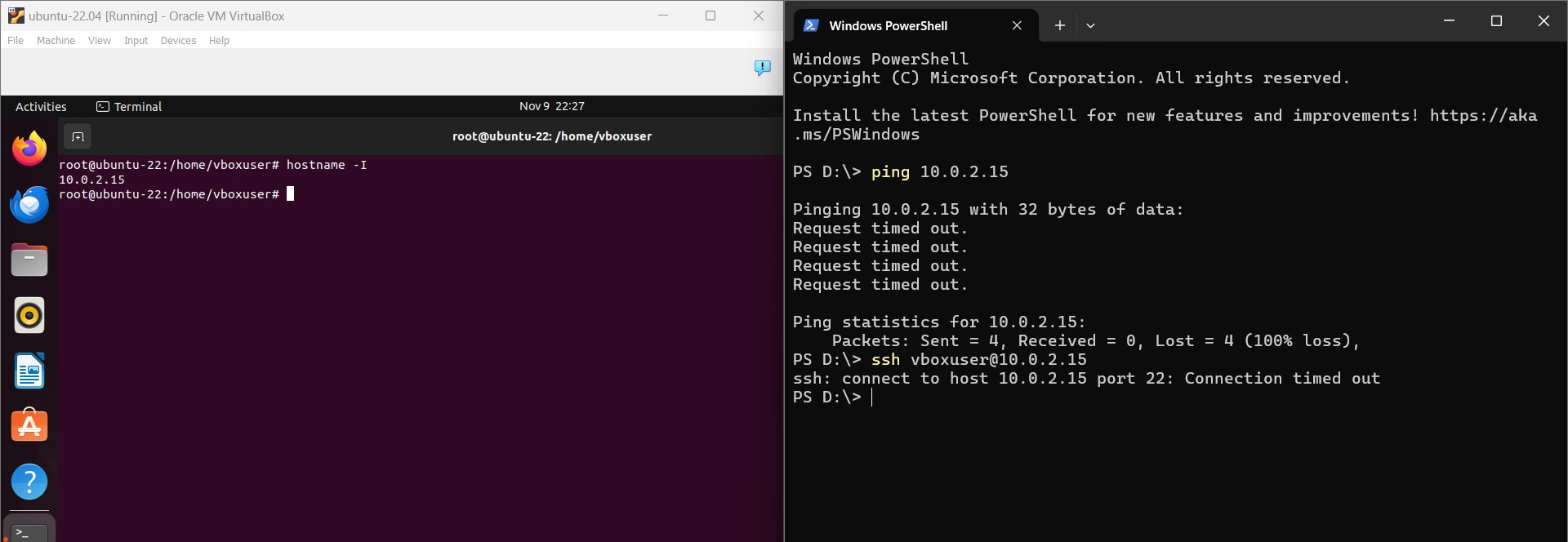
HTTPS sites may not load because they require port 443. To access such site we have to accept port 443 as well and DNS resolution has to take place before that’s why we also have to accept port 53.



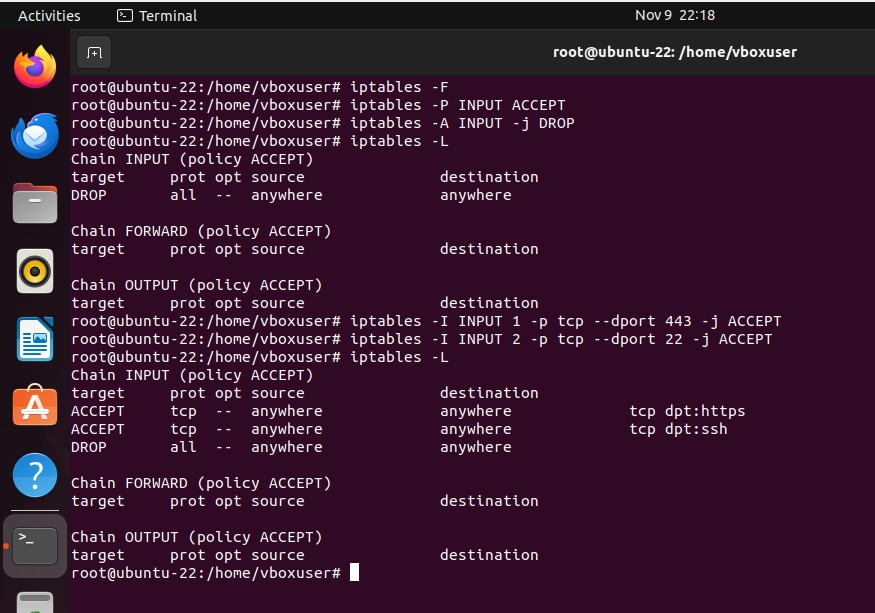
Q3) The previous question permitted only incoming packets related to web traffic. Do an insertion to the rules in the INPUT chain to permit SSH traffic. Show that you can connect to the SSH server running on the Ubuntu VM by connecting to it from another VM (centos or anything) or from the physical host machine (Windows). Include appropriate screenshots. You can get the IP address of a Linux machine by running the ifconfig command in the terminal. Refer to the screenshots (for example, under scenarios S5, S8) in the tutorial to see how you could SSH to a machine under a particular username.

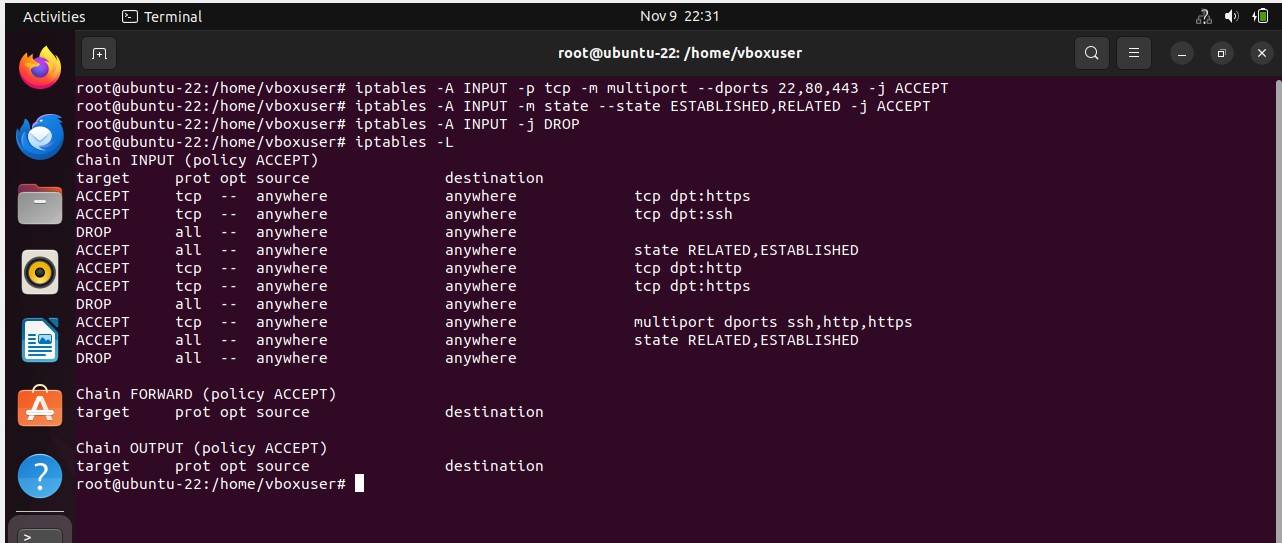


Ssh vm from remote machine

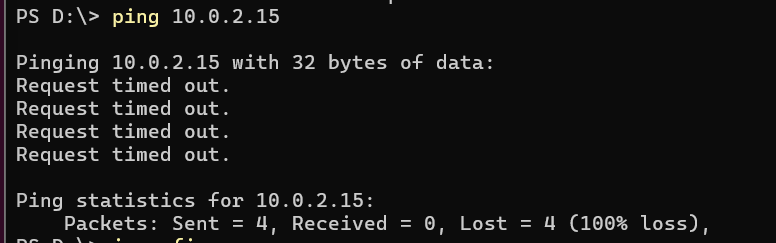


Q4) Configure your IPtables filter table on your Ubuntu VM such that sessions/packet exchange originating from the Ubuntu VM (as the source) are successful; on the other hand, sessions/packet exchange originating from a remote machine to the Ubuntu VM (as the destination) are not successful. You need to implement this scenario with the minimal number of rules and policy changes, if any. Also, explain why your set of rules and policies implementing the stated scenario will work

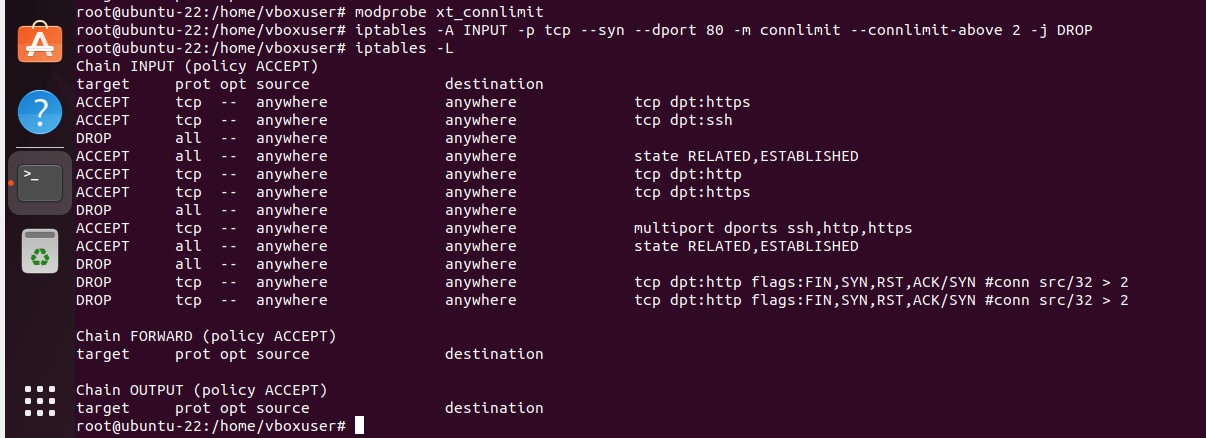




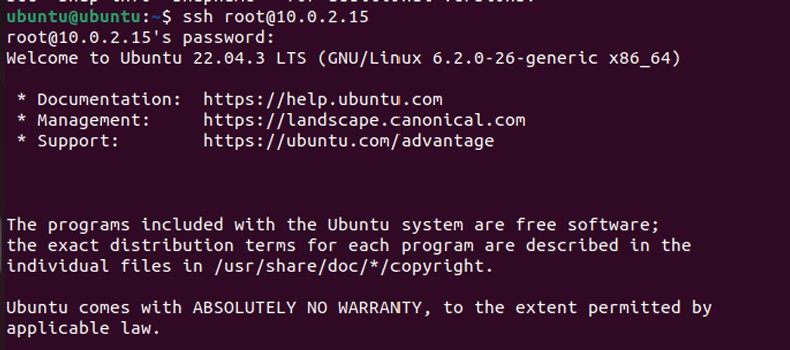
Unable to ping vm from my local machine



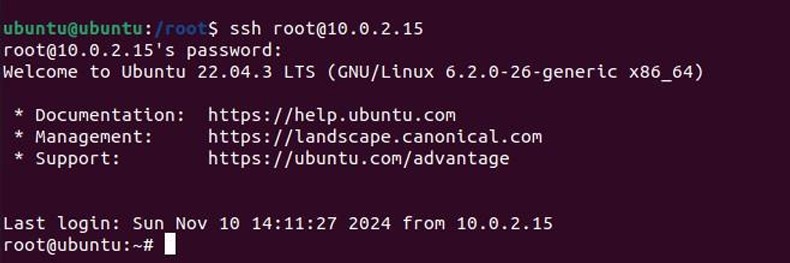
Q5) Configure your IPtables filter table to limit the number of active SSH connections to the Ubuntu VM(hosting the SSH server) is 2. Test the working of this rule by attempting to open three SSH connections, each in separate terminals, from another VM (like a CentOS VM) or from the host machine itself. Show appropriate screenshots



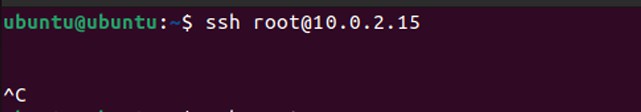
Ssh connection 1:



Ssh connection 2:



Ssh connection 3:

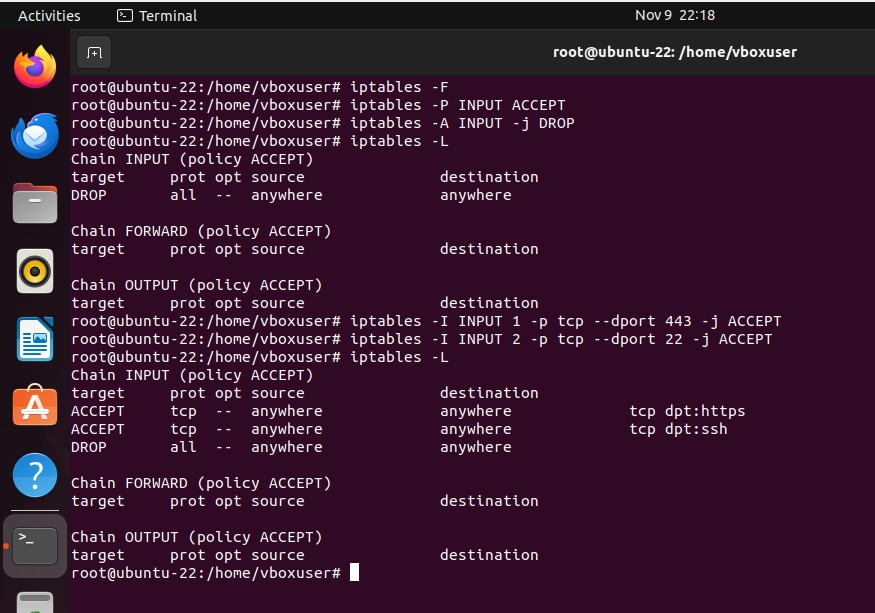


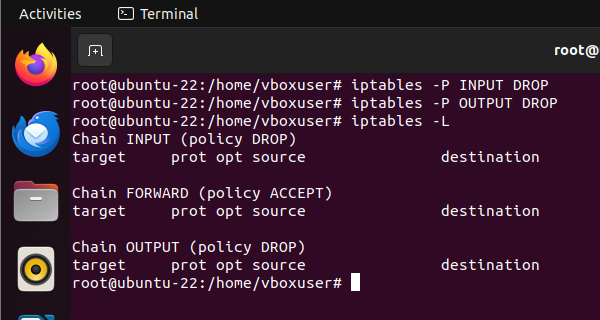
Cannot connect

Q6) Set the default policy of the INPUT and OUTPUT chains of your filter table of iptables is to DROPusing an appropriate command (show a screenshot executing the command and the output of the iptables- L command). You could use the Ubuntu VM and CentOS

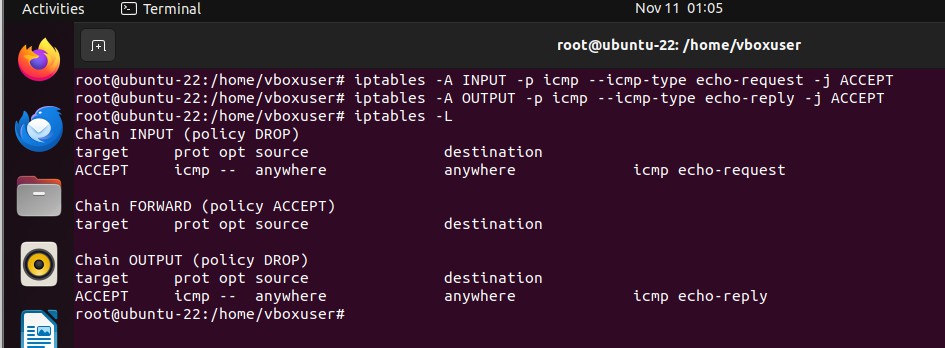
VM in your virtual environment to implement this scenario. Now configure your iptables on the Ubuntu VM to (do parts a and b independently): (a) Only allow remote machines to ping the local machine and block the local machine from pinging others. (b) Only allow the local machine to ping the remote machines and block the remote machines from pinging the local machine. (c) Allow ping communication in both directions (from the local machine to remote machine and viceversa). Note that you have to use

the--icmp-type echo-request and--icmp-type echo-reply options appropriately. Show appropriate screenshots executing the iptables commands to realize the above for (a), (b) and (c) and the structure of the iptables. Also, capture the successful or unsuccessful execution of the ping command from the local machine and remote machine (in either direction) for each of the three cases (a), (b), (c).

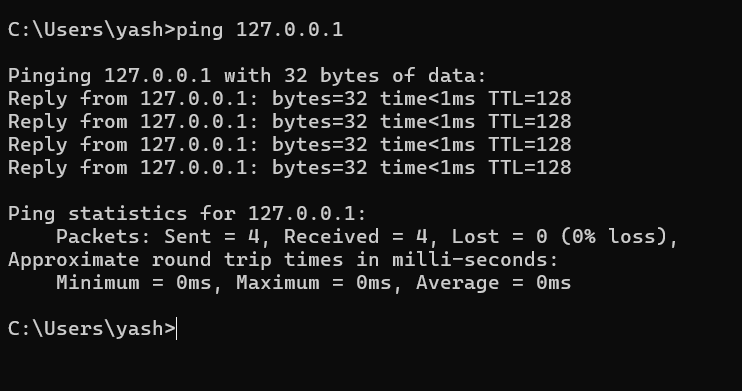




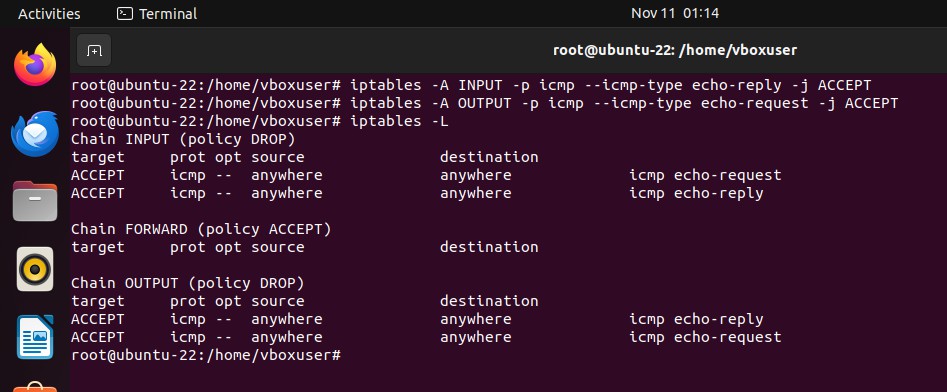
Part a:



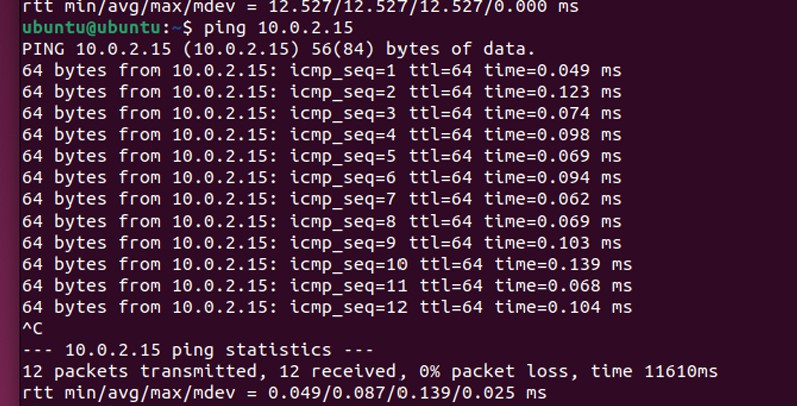
Ping vm from remote machine



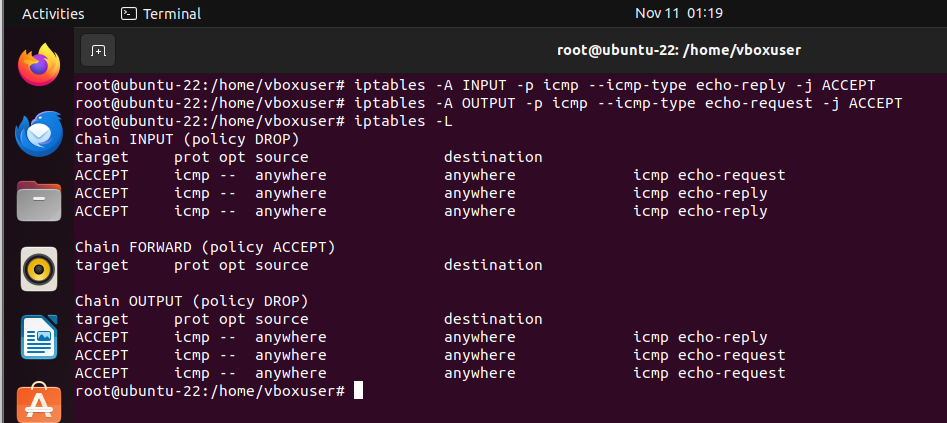
Part B:



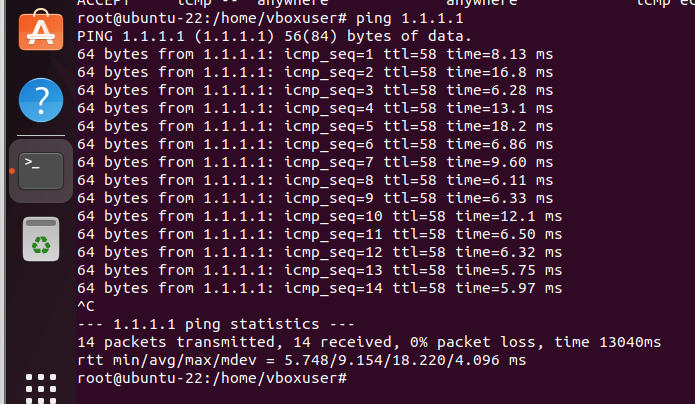
Pinging vm machine from remote machine:



Part C:



Ping another machine from VM



Ping vm from remote machine:

