ROBERT GORDON UNIVERSITY

Operating Systems Coursework

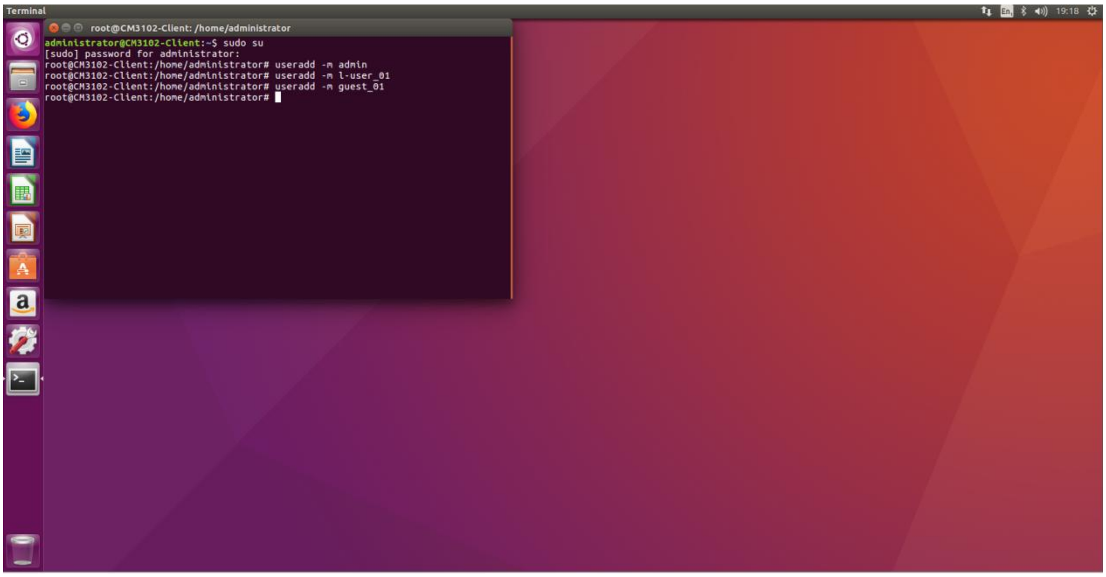
Part 1 - Setting Ubuntu virtual machines and run shell scripts

Darie-Dragos Mitoiu

10/20/2019

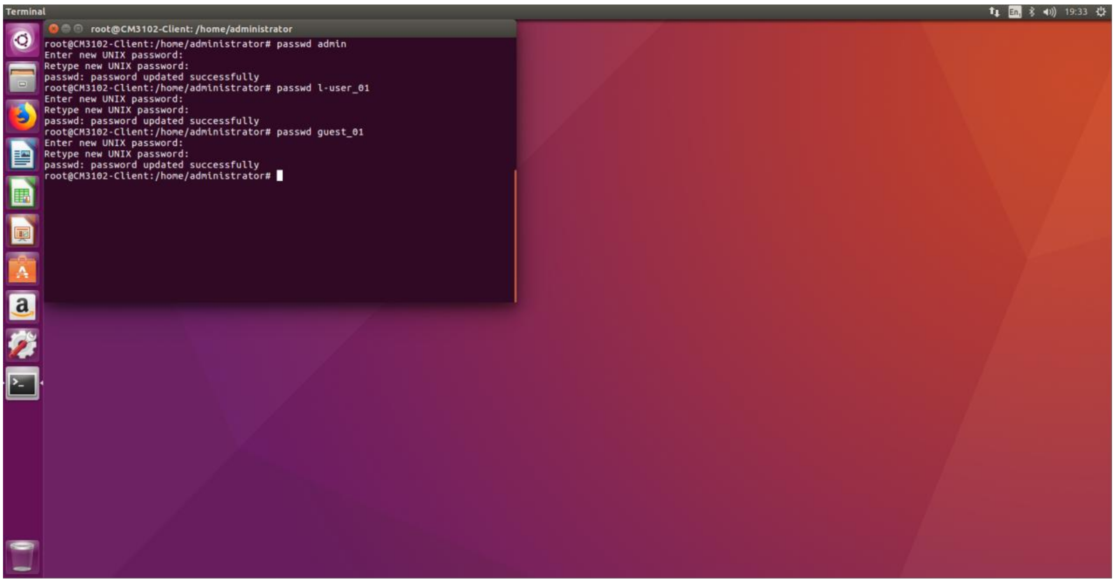
This document contains the part 1 – setting Ubuntu virtual machines and run shell scripts coursework for the operating systems module.

1.1. Create user accounts admin, l-user_01 and guest_01 that belong to the LOCAL usergroup using CLI commands:



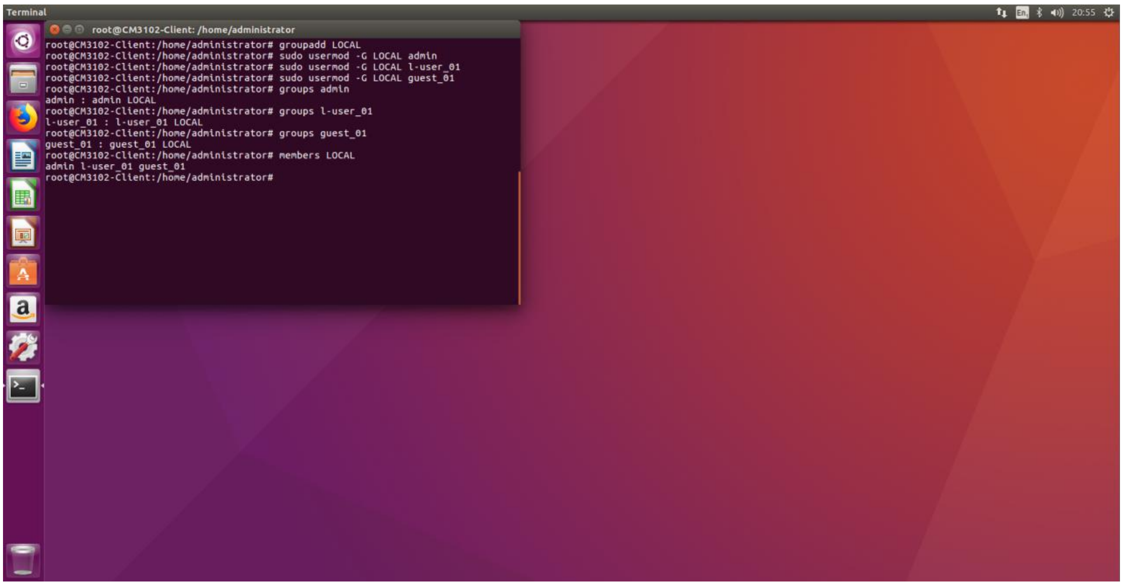
```
root@CM3102-Client: /home/administrator
administrator@CM3102-Client:~$ sudo su
[sudo] password for administrator:
root@CM3102-Client:/home/administrator# useradd -m admin
root@CM3102-Client:/home/administrator# useradd -m l-user_01
root@CM3102-Client:/home/administrator# useradd -m guest_01
root@CM3102-Client:/home/administrator# 
```

In order to create the user accounts: admin, l-user_01 and guest_01 the Ubuntu terminal was accessed and the following commands were performed in the mentioned order: “sudo su”, “useradd –m admin”, “useradd –m l-user_01”, “useradd –m guest_01”. The command “sudo su” will allow the switch of the user to the root level, this command is used in order to be able to perform the “useradd” command, the command “useradd –m admin” will allow the creation of the user called admin, the command “useradd –m l-user_01” will allow the creation of the user called l-user_01 and the command “useradd –m guest_01” will allow the creation of the user called guest_01.



```
root@CM3102-Client: /home/administrator
root@CM3102-Client:/home/administrator# passwd admin
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@CM3102-Client:/home/administrator# passwd l-user_01
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@CM3102-Client:/home/administrator# passwd guest_01
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@CM3102-Client:/home/administrator# 
```

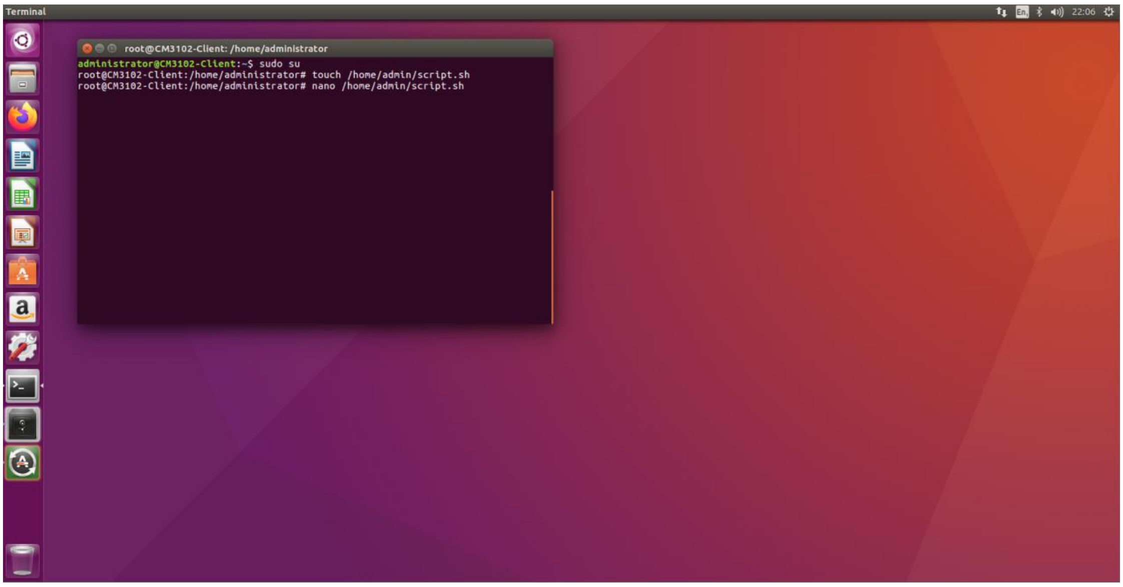
Once the user accounts: admin, l-user_01 and guest_01 have been created, each user account must have a password in order to be used, in order to set a password for each account the following commands were performed in the mentioned order: “passwd admin”, “passwd l-user_01” and “passwd guest_01”. After each command a password had to be provided and then confirmed in order to be associated with the mentioned user account. The command “passwd admin” will allow the creation of a password for the user called admin, the command “passwd l-user_01” will allow the creation of a password for the user called l-user_01 and the command “passwd guest_01” will allow the creation of a password for the user guest_01.



```
root@CM3102-Client:/home/administrator# groupadd LOCAL
root@CM3102-Client:/home/administrator# sudo usermod -G LOCAL admin
root@CM3102-Client:/home/administrator# sudo usermod -G LOCAL l-user_01
root@CM3102-Client:/home/administrator# sudo usermod -G LOCAL guest_01
root@CM3102-Client:/home/administrator# groups admin
admin : admin LOCAL
root@CM3102-Client:/home/administrator# groups l-user_01
l-user_01 : l-user_01 LOCAL
root@CM3102-Client:/home/administrator# groups guest_01
guest_01 : guest_01 LOCAL
root@CM3102-Client:/home/administrator# members LOCAL
admin l-user_01 guest_01
root@CM3102-Client:/home/administrator#
```

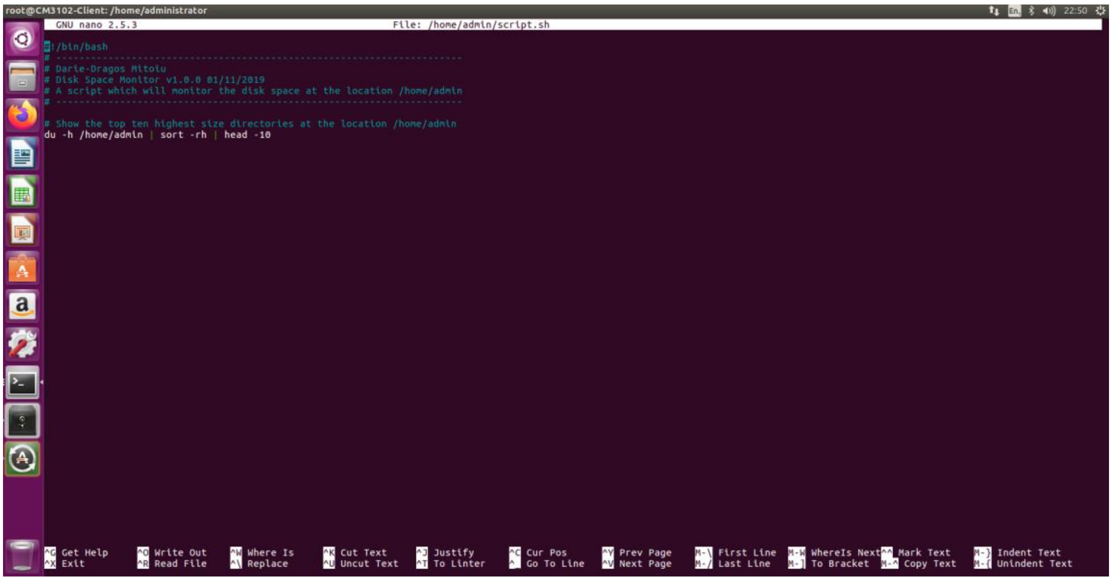
In order to add the user accounts: admin, l-user_01 and guest_01 to the group "LOCAL" the following commands were performed in the mentioned order: "groupadd LOCAL", "sudo usermod –G LOCAL admin", "sudo usermod –G LOCAL l-user_01", "sudo usermod –G LOCAL guest_01", "groups admin", "groups l-user_01", "groups guest_01". The command "groupadd LOCAL" will allow the creation of the group called "LOCAL", the command "sudo usermod –G LOCAL admin" will add the user admin to the "LOCAL" group, the command "sudo usermod –G l-user_01" will add the user l-user_01 to the "LOCAL" group, the command "sudo usermod –g guest_01" will add the user guest_01 to the "LOCAL" group and last but not the least to verify if the users belong to that group the command "groups" associated with the user account will show this information.

1.2. In the admin home directory write a shell script that monitors disk space, namely the biggest top ten directories (N.B. use the du command with appropriate options):



```
administrator@CM3102-Client:~$ sudo su
root@CM3102-Client:/home/administrator# touch /home/admin/script.sh
root@CM3102-Client:/home/administrator# nano /home/admin/script.sh
```

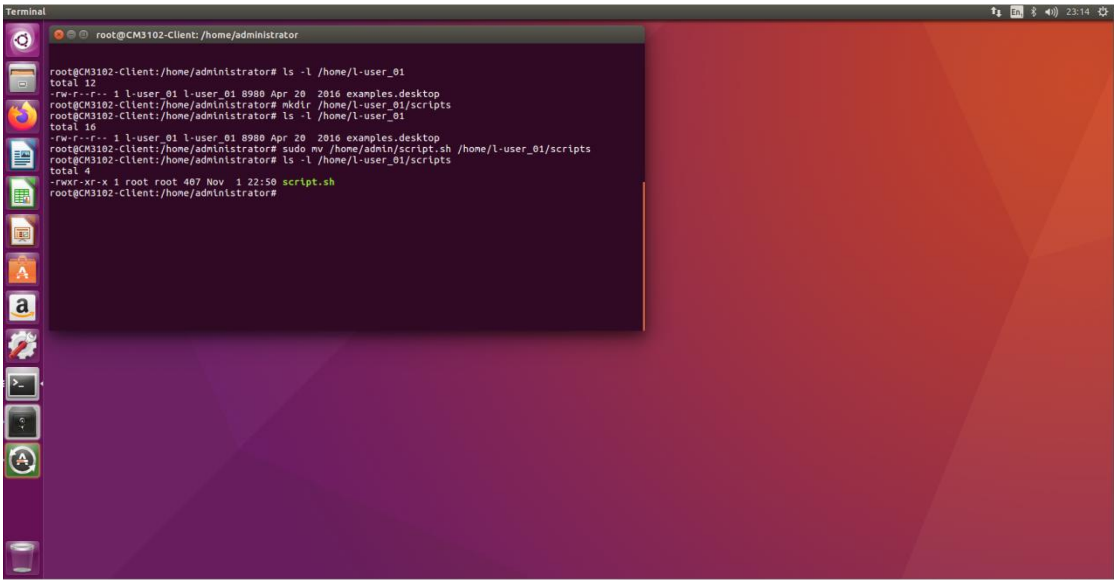
In order to write a shell script that monitors disk space, namely the biggest top ten directories at the location /home/admin the following commands must be executed: "sudo su", "touch /home/admin/script.sh", "nano /home/admin/script.sh". The command "sudo su" will allow the switch of the user to the root level where the next steps will be performed, the command "touch /home/admin/script.sh" will allow the creation of the file called "script.sh" at the location /home/admin, the command "nano /home/admin/script.sh" will allow the file called script.sh to be opened using the command line text editor called nano, the nano text editor will be used to write the script, as we can see in the above image the last command was not executed, is there just for information purpose only.



```
  GNU nano 2.5.3                                       File: /home/admin/script.sh

#!/bin/bash
# ----------------------------------------------------------------------
# Darie-Dragos Mitoiu
# Disk Space Monitor v1.0.0 01/11/2019
# A script which will monitor the disk space at the location /home/admin
# ----------------------------------------------------------------------

# Show the top ten highest size directories at the location /home/admin
du -h /home/admin | sort -rh | head -10
```

In order to visualise the top ten highest size directories at the location /home/admin, the following command must be executed: “du –h /home/admin | sort –rh | head -10”. The command “du –h /home/admin | sort –rh | head -10” will allow the visualisation of the disk usage in human readable mode at the location /home/admin, sorting the files based on the size and allowing only 10 directories to be shown once the command is executed. As we saw in the previous image the nano command line editor was used in order to write the script, we can see the content of the script in the above image, which contains the standard comments and the command that will allow the visualisation of the disk usage.

1.3. Store the script file in the /scripts subdirectory, owned by the l-user_01:



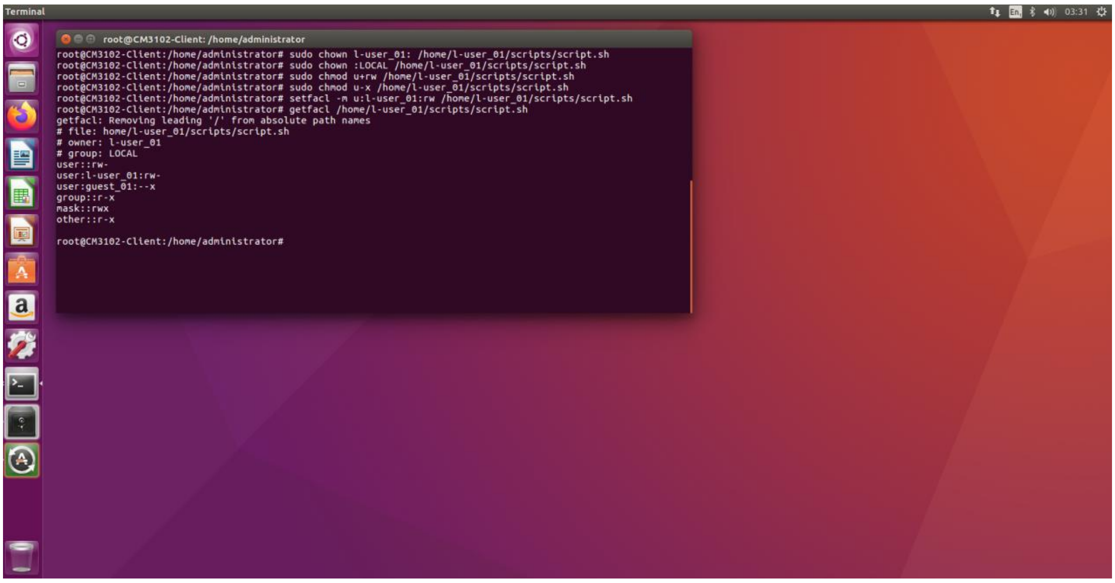
```
root@CM3102-Client:/home/administrator# ls -l /home/l-user_01
total 12
-rw-r--r-- 1 l-user_01 l-user_01 8980 Apr 20  2016 examples.desktop
root@CM3102-Client:/home/administrator# mkdir /home/l-user_01/scripts
root@CM3102-Client:/home/administrator# ls -l /home/l-user_01
total 16
-rw-r--r-- 1 l-user_01 l-user_01 8980 Apr 20  2016 examples.desktop
root@CM3102-Client:/home/administrator# sudo mv /home/admin/script.sh /home/l-user_01/scripts
root@CM3102-Client:/home/administrator# ls -l /home/l-user_01/scripts
total 4
-rwxr-xr-x 1 root root 407 Nov  1 22:50 script.sh
root@CM3102-Client:/home/administrator#
```

In order to store the script file in the /scripts subdirectory, owned by the l-user_01, the following commands must be executed: “ls –l /home/l-user_01”, “mkdir /home/l-user_01/scripts”, “ls –l /home/l-user_01”, “sudo mv /home/admin/script.sh /home/l-user_01/scripts”, “ls –l /home/l-user_01/scripts”. The command “ls –l /home/l-user_01” will allow the visualisation in a long list of the home directory of the l-user_01, the command “mkdir /home/l-user_01/scripts” will create a new directory called scripts at the location

/home/l-user_01, the command “ls –l /home/l-user_01” will verify if the directory was created, the command “mv /home/admin/script.sh /home/l-user_01/scripts” will move the script file from the home directory of the admin user to the scripts directory owned by the l-user_01 and the command “ls –l /home/l-user_01/scripts” will allow the verification of the presence of the script file in the scripts directory.

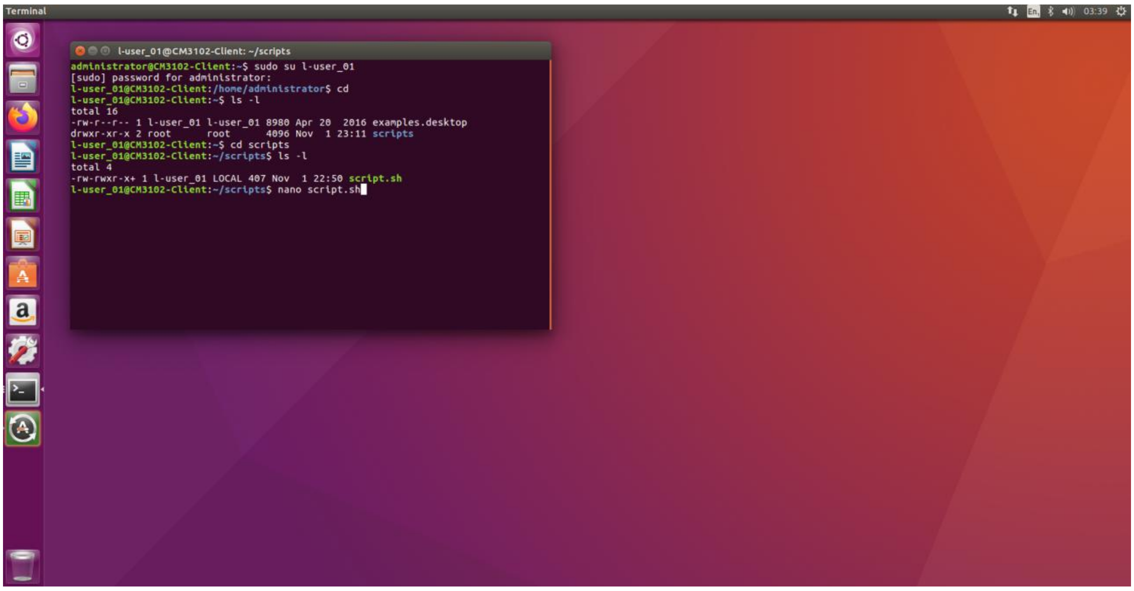
1.4. Using both chmod command and access control list (ACLs):

Set the write permission to l-user_01. Check that l-user_01 can exercise this permission by modifying the script so that it can format the output of the script.



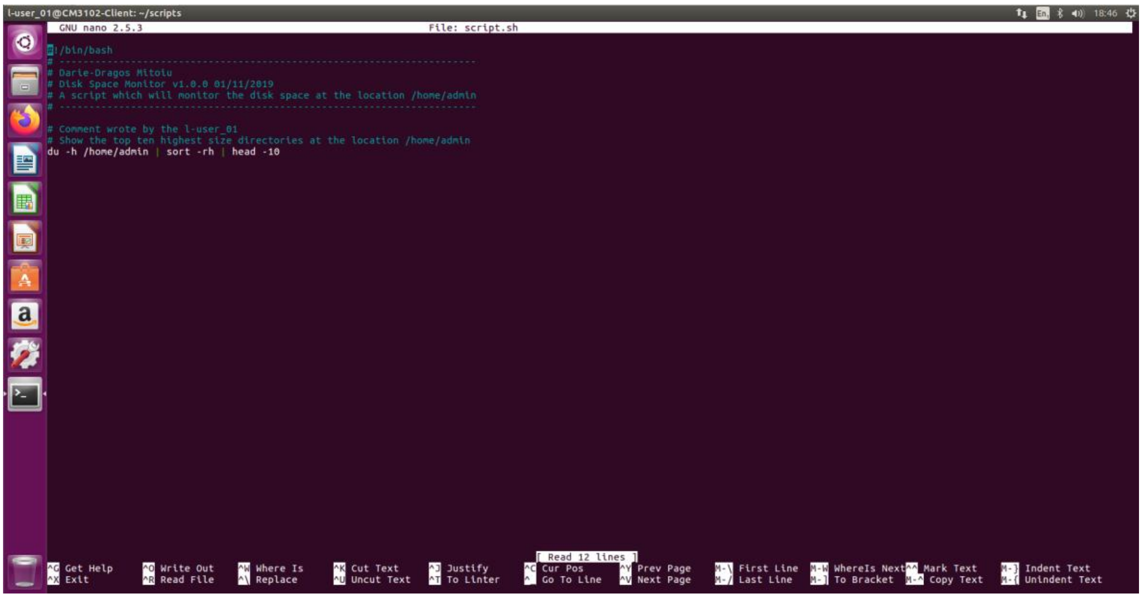
```
root@CM3102-Client: /home/administrator
root@CM3102-Client:/home/administrator# sudo chown l-user_01: /home/l-user_01/scripts/script.sh
root@CM3102-Client:/home/administrator# sudo chown :LOCAL /home/l-user_01/scripts/script.sh
root@CM3102-Client:/home/administrator# sudo chmod u+rw /home/l-user_01/scripts/script.sh
root@CM3102-Client:/home/administrator# sudo chmod u-x /home/l-user_01/scripts/script.sh
root@CM3102-Client:/home/administrator# setfacl -m u:l-user_01:rw /home/l-user_01/scripts/script.sh
root@CM3102-Client:/home/administrator# getfacl /home/l-user_01/scripts/script.sh
getfacl: Removing leading '/' from absolute path names
# file: home/l-user_01/scripts/script.sh
# owner: l-user_01
# group: LOCAL
user::rw-
user:l-user_01:rw-
user:guest_01:--x
group::r-x
mask::rwx
other::r-x

root@CM3102-Client:/home/administrator#
```

In order to set the write permission to the l-user_01 using the chmod command and access control list, the following commands must be executed: “sudo chown l-user_01: /home/l-user_01/scripts/script.sh”, “sudo chown :LOCAL /home/l-user_01/scripts/script.sh”, “sudo chmod u+rw /home/l-user_01/scripts/script.sh”, “sudo chmod u-x /home/l-user_01/scripts/script.sh”, “setfacl –m u:l-user_01:w /home/l-user_01/scripts/script.sh” and “getfacl /home/l-user_01/scripts/script.sh”. The command “sudo chown l-user_01: /home/l-user_01/scripts/script.sh” will change the script.sh file ownership to the l-user_01, in order to set it using the chmod command, the command “sudo chown :LOCAL /home/l-user_01/scripts/script.sh” will change the script.sh file group ownership to the LOCAL group which will allow the chmod command to be used to the guest_01 in the following section, the command “sudo chmod u+w /home/l-user_01/scripts/script.sh” will add the write permission to the owner type of user which was set to l-user_01, the command “sudo chmod u-rx /home/scripts/script.sh” will remove the read and execute permissions for the l-user_01 as they were not mentioned in the requirement, only the write permission was mentioned, the command “setfacl –m u:l-user_01:w /home/l-user_01/scripts/script.sh” will give write permission to the l-user_01 and the command “getfacl /home/l-user_01/scripts/script.sh” will show the file and user associated with their permissions.



```
administrator@CM3102-Client:~$ sudo su l-user_01
[sudo] password for administrator:
l-user_01@CM3102-Client:/home/administrator$ cd
l-user_01@CM3102-Client:~$ ls -l
total 16
-rw-r--r-- 1 l-user_01 l-user_01 8980 Apr 20  2016 examples.desktop
drwxr-xr-x 2 root      root      4096 Nov  1 23:11 scripts
l-user_01@CM3102-Client:~$ cd scripts
l-user_01@CM3102-Client:~/scripts$ ls -l
total 4
-rw-rwxr-x+ 1 l-user_01 LOCAL 407 Nov  1 22:50 script.sh
l-user_01@CM3102-Client:~/scripts$ nano script.sh
```

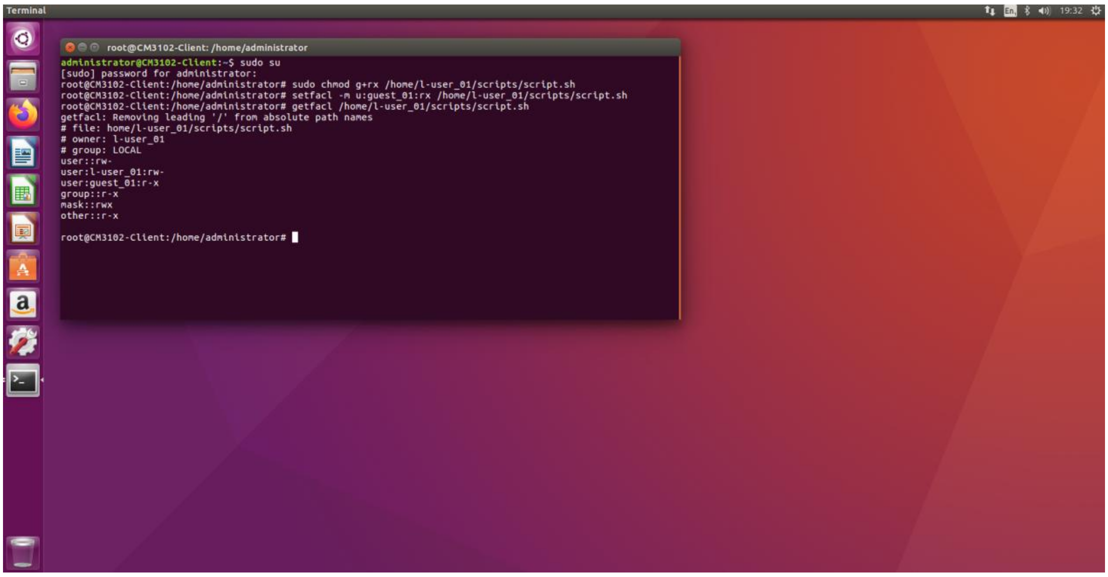
In order to verify if the l-user_01 has write permission, the following commands must be executed: “sudo su l-user_01”, “cd”, “ls -l”, “cd scripts”, “ls -l”, “nano script.sh”. The command “sudo su l-user_01” will allow the access to the l-user_01 account, the command “cd” will allow us to leave the /home/administrator location, the command “ls -l” will list the files/directories of the /home/l-user_01/ location in a long list, the command “cd scripts” will access the scripts directory, the command “ls -l” will show the files/directories in a long list and the command “nano script.sh” which is not executed in the image from above will allow the script.sh file to be opened and modified since the account l-user_01 has both read and write permissions, the command will be executed in the image from below.



```
  GNU nano 2.5.3                                             File: script.sh

#!/bin/bash
# -------------------------------------------------------------------------
# Darie-Dragos Mitoiu
# Disk Space Monitor v1.0.0 01/11/2019
# A script which will monitor the disk space at the location /home/admin
# -------------------------------------------------------------------------

# Comment wrote by the l-user_01
# Show the top ten highest size directories at the location /home/admin
du -h /home/admin | sort -rh | head -10

                                          [ Read 12 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    ^Y Prev Page  M-\ First Line M-W WhereIs Next^^ Mark Text   M-} Indent Text
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Linter  ^_ Go To Line ^V Next Page  M-/ Last Line  M-] To Bracket M-^ Copy Text  M-{ Unindent Text
```

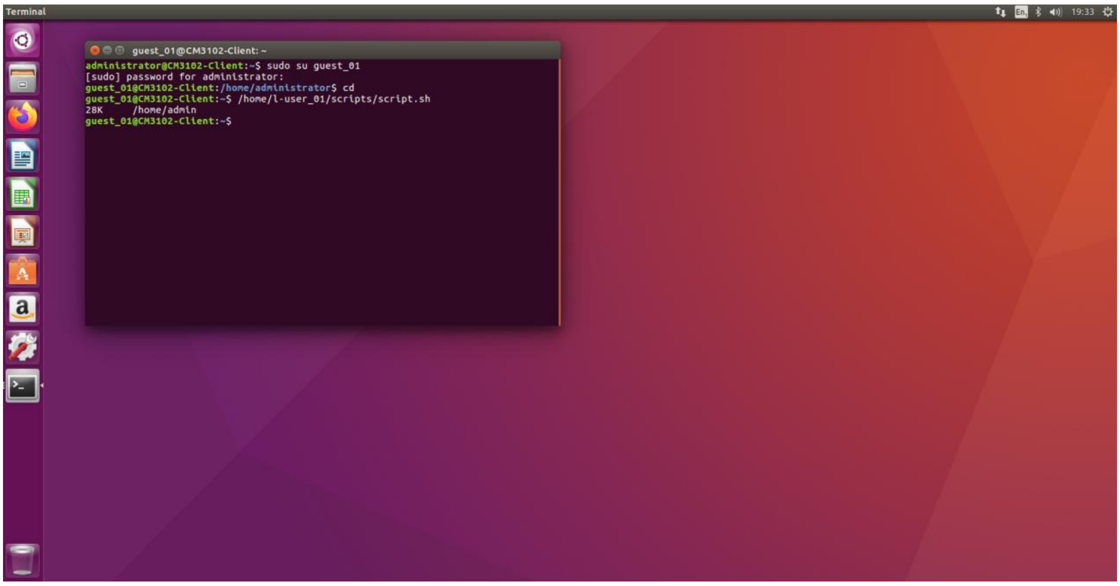
In order to verify if the l-user_01 has write permission for the script.sh file located at the location /home/l-user_01/scripts/, the command “nano script.sh” was executed. The command “nano script.sh” will open the file in the command line editor nano, which will allow the file to be read since the l-user_01 has also read permission in order to visualise the content of the file and place new text in the file at the right position, in order to verify if the l-user_01 has write permission a comment has been added to the script and then the file was closed using the shortcut ctrl+x following the saving of the file, now we can confirm that the user has write permission for the script.sh file.

Set the execute permission to guest_01. Check that guest_01 can run the modified script.



```
administrator@CM3102-Client:~$ sudo su
[sudo] password for administrator:
root@CM3102-Client:/home/administrator# sudo chmod g+rx /home/l-user_01/scripts/script.sh
root@CM3102-Client:/home/administrator# setfacl -m u:guest_01:rx /home/l-user_01/scripts/script.sh
root@CM3102-Client:/home/administrator# getfacl /home/l-user_01/scripts/script.sh
getfacl: Removing leading '/' from absolute path names
# file: home/l-user_01/scripts/script.sh
# owner: l-user_01
# group: LOCAL
user::rw-
user:l-user_01:rw-
user:guest_01:r-x
group::r-x
mask::rwx
other::r-x

root@CM3102-Client:/home/administrator# 
```

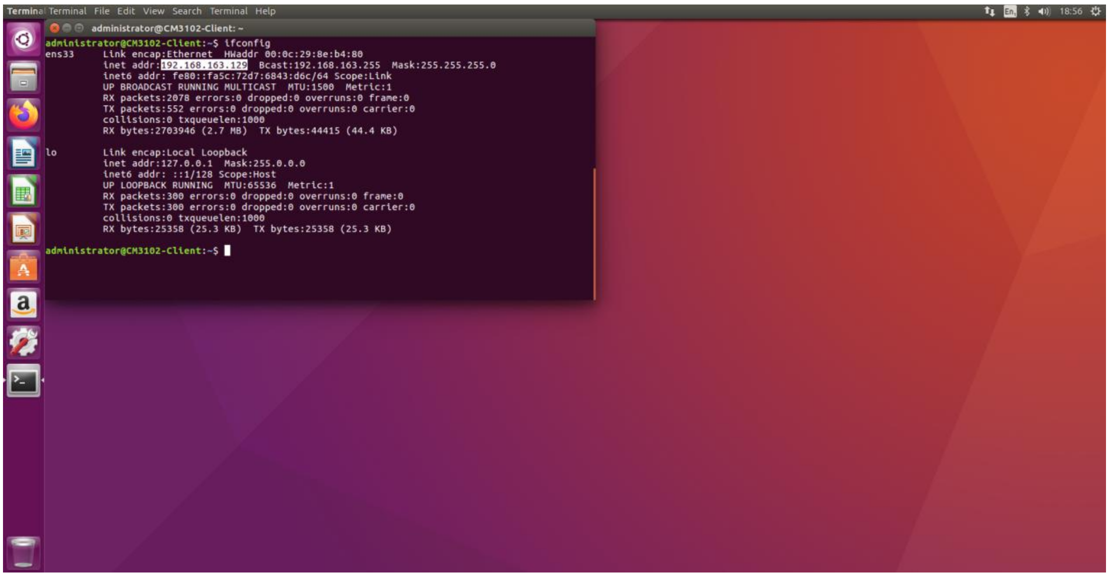
In order set the execute permission to guest_01, the following commands must be executed: “sudo chmod g+rx /home/l-user_01/scripts/script.sh”, “setfacl –m u:guest_01:rx /home/l-user_01/scripts/script.sh”, “getfacl /home/l-user_01/scripts/script.sh”. The command “sudo chmod g+rx /home/l-user_01/scripts/script.sh” will add the execute permission to the LOCAL group, the guest_01 is part of the LOCAL group so it will be able to execute the file, this command has been used because the l-user_01 is already the owner of the file, the command “setfacl –m u:guest_01:rx /home/l-user_01/scripts/script.sh” will allow the guest user to execute the script.sh file and the command “getfacl /home/l-user_01/scripts/script.sh” will show the users and the permissions over the script.sh file.



```
administrator@CM3102-Client:~$ sudo su guest_01
[sudo] password for administrator:
guest_01@CM3102-Client:/home/administrator$ cd
guest_01@CM3102-Client:~$ /home/l-user_01/scripts/script.sh
28K     /home/admin
guest_01@CM3102-Client:~$
```

In order to verify if the guest_01 account has execute permission the following commands must be executed: “sudo su guest_01”, “cd”, “/home/l-user_01/scripts/script.sh”. The command “sudo su guest_01” will allow the access of the account called gust_01, once this command is executed the current user password has to be provided and then the operation will be completed, the command “cd” will change the current location to the /home/guest_01, the command /home/l-user_01/scripts/script.sh will execute the script called script.sh and then the result of the script will be printed below. As we can see in the image from above the guest_01 has execute permission over the /home/l-user_01/scripts/script.sh file.

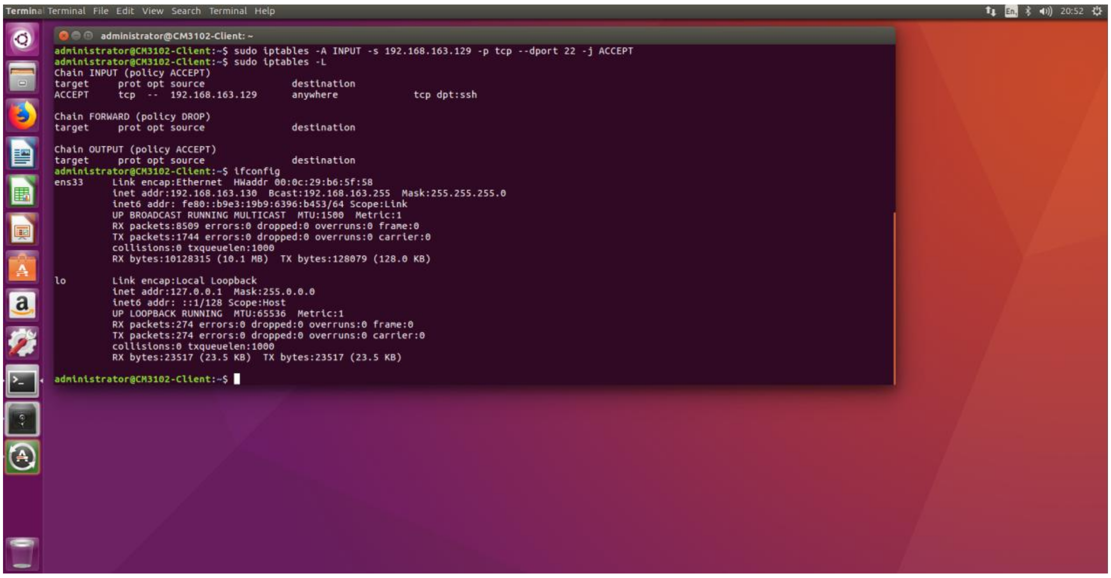
2.1. Configure a firewall on CM3102_Serve_VM so that secure shell server is enabled:



Terminal Terminal File Edit View Search Terminal Help
18:56
administrator@CM3102-Client: ~

```
administrator@CM3102-Client:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:8e:b4:80
          inet addr:192.168.163.129  Bcast:192.168.163.255  Mask:255.255.255.0
          inet6 addr: fe80::fa5c:72d7:6843:d6c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2078 errors:0 dropped:0 overruns:0 frame:0
          TX packets:552 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2703946 (2.7 MB)  TX bytes:44415 (44.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:300 errors:0 dropped:0 overruns:0 frame:0
          TX packets:300 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:25358 (25.3 KB)  TX bytes:25358 (25.3 KB)

administrator@CM3102-Client:~$
```

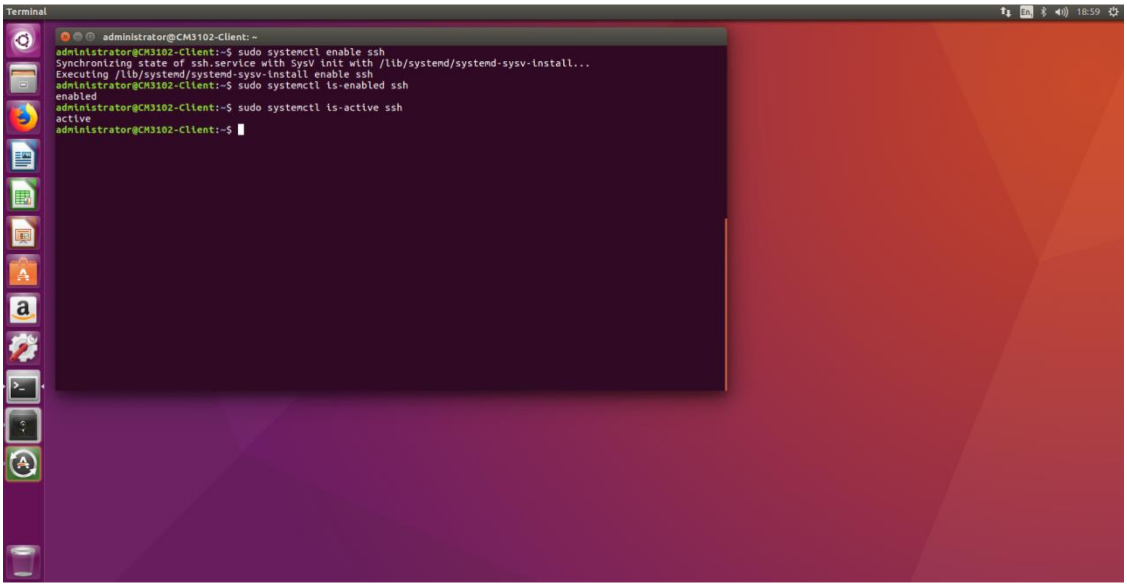
In order to configure a firewall on CM3102_Serve_VM so that secure shell server is enabled, the IP address of the client virtual machine must be discovered and then used in the server virtual machine as a valid address for incoming connections towards the server. In order to find out the IP address of the client virtual machine the following command must be executed: “ifconfig”. The “ifconfig” command will show the client’s virtual machine IP address which can be seen in the above image. The IP address is 192.168.163.129, this IP address will be used in the next steps in the image below.



Terminal Terminal File Edit View Search Terminal Help
20:52
administrator@CM3102-Client: ~

```
administrator@CM3102-Client:~$ sudo iptables -A INPUT -s 192.168.163.129 -p tcp --dport 22 -j ACCEPT
administrator@CM3102-Client:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  192.168.163.129      anywhere             tcp dpt:ssh

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
administrator@CM3102-Client:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:b6:5f:58
          inet addr:192.168.163.130  Bcast:192.168.163.255  Mask:255.255.255.0
          inet6 addr: fe80::b9e3:19b9:6396:b453/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8509 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1744 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10128315 (10.1 MB)  TX bytes:128079 (128.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:274 errors:0 dropped:0 overruns:0 frame:0
          TX packets:274 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23517 (23.5 KB)  TX bytes:23517 (23.5 KB)

administrator@CM3102-Client:~$
```

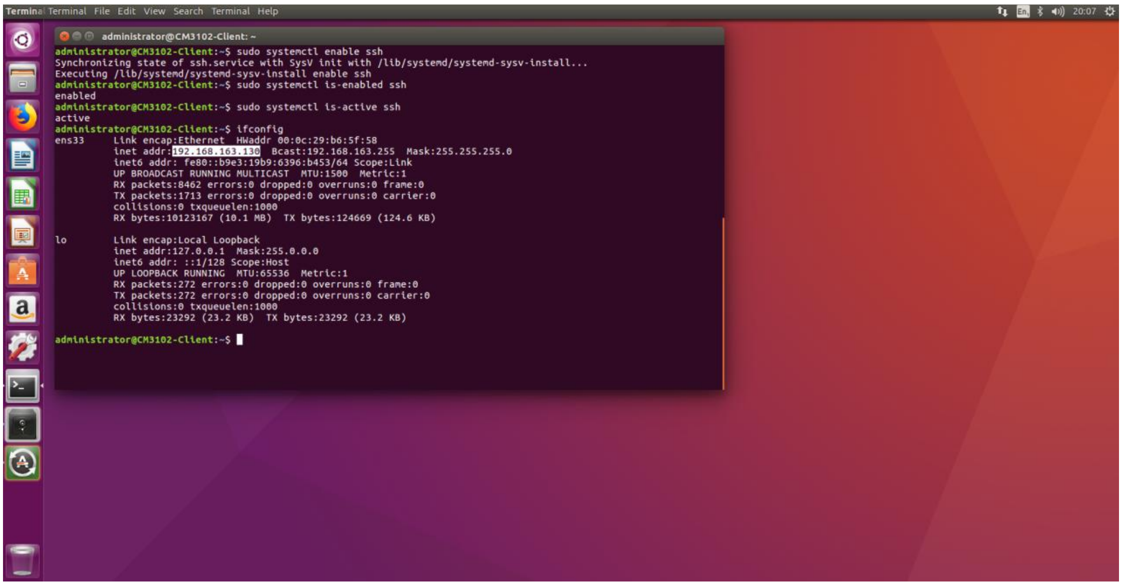
In order to add a firewall rule for the an incoming connection from the client virtual machine the following commands must be executed: “sudo iptables -L”, “sudo iptables –A INPUT –s 192.168.163.129 –p tcp –dport 22 –j ACCEPT”, “sudo iptables -L”. The command “sudo iptables -L” will show the firewall rules set for the server virtual machine, as we can see in the image above there are no rules set, the command “sudo iptables –A INPUT –s 192.168.163.129 –p tcp –dport 22 –j ACCEPT” will add a new rule for the incoming connections from the 192.168.163.129 IP address, this rule will add the IP mentioned in the accepted IP’s for the incoming connections towards the server virtual machine, the command “sudo iptables -L” executed after the addition of the rule will show the rule just added.



```
administrator@CM3102-Client: ~
administrator@CM3102-Client:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable ssh
administrator@CM3102-Client:~$ sudo systemctl is-enabled ssh
enabled
administrator@CM3102-Client:~$ sudo systemctl is-active ssh
active
administrator@CM3102-Client:~$ 
```

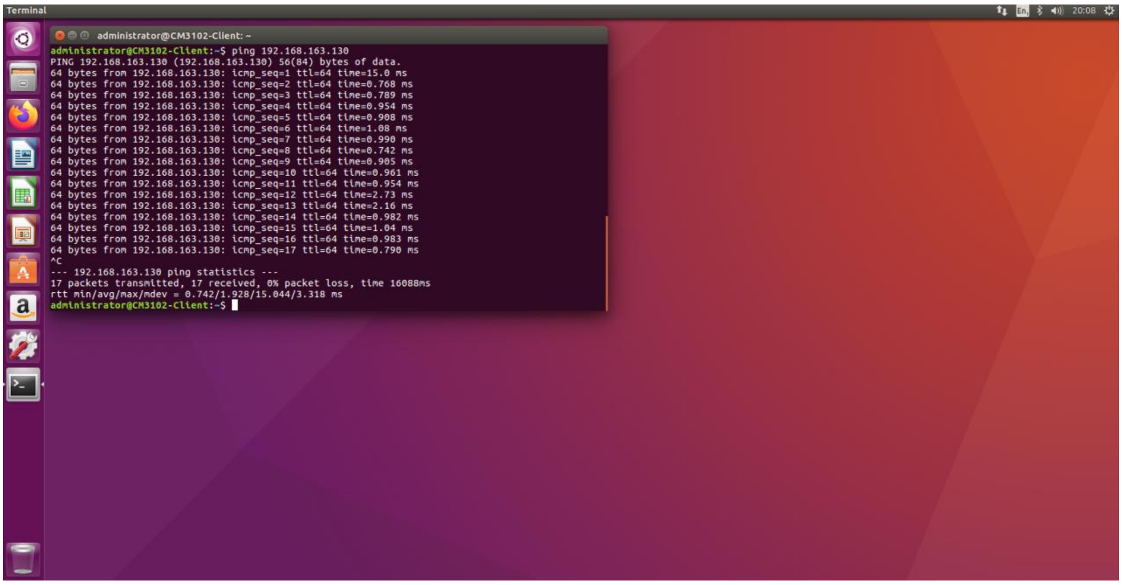
In order to ensure that the secure shell server is enabled the following commands must be executed: “sudo systemctl enable ssh”, “sudo systemctl is-enabled ssh”, “sudo systemctl is-active ssh”. The command “sudo systemctl enable ssh” will enable the secure shell server so the client virtual machine will have the possibility to connect in the next steps, the command “sudo systemctl is-enabled ssh” will show if the secure shell server is enabled, the command “sudo systemctl is-active ssh” will show if the secure shell server is active. The results from above image will show that the secure shell server was enabled successfully and is active at the same time.

2.2. Ensure that CM3102_Client_VM can connect to the CM3102_Serve_VM virtual machine – use the ping command to check connectivity:



```
administrator@CM3102-Client: ~
administrator@CM3102-Client:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable ssh
administrator@CM3102-Client:~$ sudo systemctl is-enabled ssh
enabled
administrator@CM3102-Client:~$ sudo systemctl is-active ssh
active
administrator@CM3102-Client:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:b6:5f:58
          inet addr:192.168.163.130  Bcast:192.168.163.255  Mask:255.255.255.0
          inet6 addr: fe80::b9e3:19b9:6396:b453/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8462 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1713 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10123167 (10.1 MB)  TX bytes:124669 (124.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:272 errors:0 dropped:0 overruns:0 frame:0
          TX packets:272 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23292 (23.2 KB)  TX bytes:23292 (23.2 KB)

administrator@CM3102-Client:~$ 
```

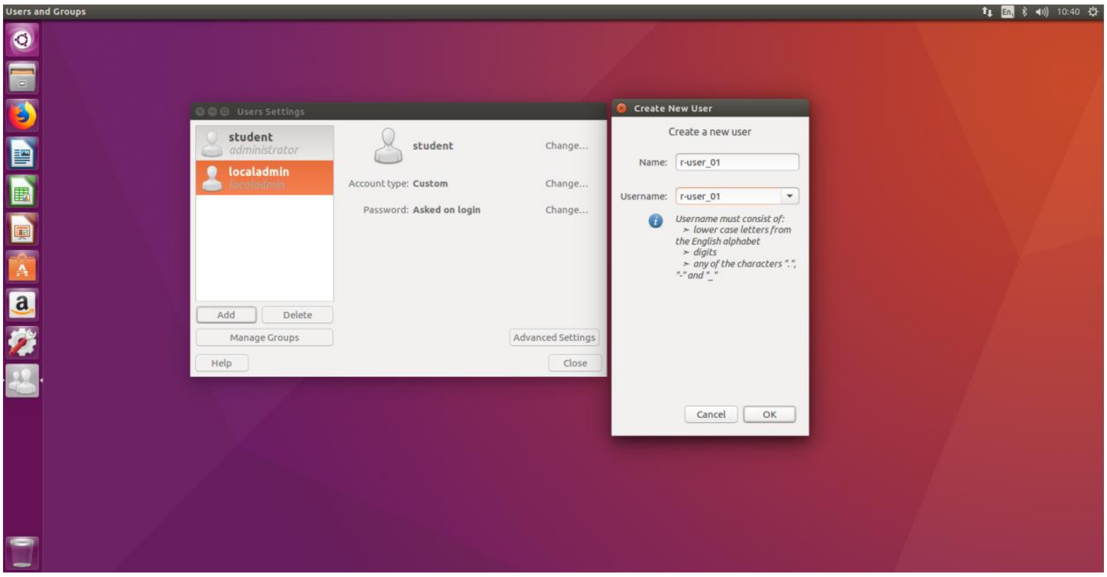
In order to ensure that the CM3102_Client_VM can connect to the CM3102_Serve_VM virtual machine, the first step will be to find out the IP address of the server virtual machine in order to use it in the client virtual machine. In order to get the IP address of the server virtual machine the following command must be executed: “ifconfig”, the command “ifconfig” will show the IP address of the server virtual machine in the above image. The IP address of the server virtual machine is: 192.168.163.130 and the IP address of the client virtual machine is 192.168.163.129 as seen in the previous images.



```
administrator@CM3102-Client:~$ ping 192.168.163.130
PING 192.168.163.130 (192.168.163.130) 56(84) bytes of data.
64 bytes from 192.168.163.130: icmp_seq=1 ttl=64 time=15.0 ms
64 bytes from 192.168.163.130: icmp_seq=2 ttl=64 time=0.768 ms
64 bytes from 192.168.163.130: icmp_seq=3 ttl=64 time=0.789 ms
64 bytes from 192.168.163.130: icmp_seq=4 ttl=64 time=0.954 ms
64 bytes from 192.168.163.130: icmp_seq=5 ttl=64 time=0.908 ms
64 bytes from 192.168.163.130: icmp_seq=6 ttl=64 time=1.08 ms
64 bytes from 192.168.163.130: icmp_seq=7 ttl=64 time=0.990 ms
64 bytes from 192.168.163.130: icmp_seq=8 ttl=64 time=0.742 ms
64 bytes from 192.168.163.130: icmp_seq=9 ttl=64 time=0.905 ms
64 bytes from 192.168.163.130: icmp_seq=10 ttl=64 time=0.961 ms
64 bytes from 192.168.163.130: icmp_seq=11 ttl=64 time=0.954 ms
64 bytes from 192.168.163.130: icmp_seq=12 ttl=64 time=2.73 ms
64 bytes from 192.168.163.130: icmp_seq=13 ttl=64 time=2.16 ms
64 bytes from 192.168.163.130: icmp_seq=14 ttl=64 time=0.982 ms
64 bytes from 192.168.163.130: icmp_seq=15 ttl=64 time=1.04 ms
64 bytes from 192.168.163.130: icmp_seq=16 ttl=64 time=0.983 ms
64 bytes from 192.168.163.130: icmp_seq=17 ttl=64 time=0.790 ms
^C
--- 192.168.163.130 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16088ms
rtt min/avg/max/mdev = 0.742/1.928/15.044/3.318 ms
administrator@CM3102-Client:~$
```

In order to verify the connectivity between the client virtual machine and the server virtual machine the following command must be executed: “ping 192.168.163.130”. The command “ping 192.168.163.130” will verify the connectivity between the client virtual machine and the server virtual machine, the IP address used with the ping command is the IP address of the server virtual machine and the command was executed using the client virtual machine, the result of the command is 17 packets transmitted, 17 packets received and 0% packet loss, as we can see in the image above the client virtual machine connected to the server virtual machine successfully.

2.3. Using the gnome-system-tools package create a user account r-user_01 that belongs to the REMOTE group, which also needs to be created and combines all users that can access the server.



Account type: Custom

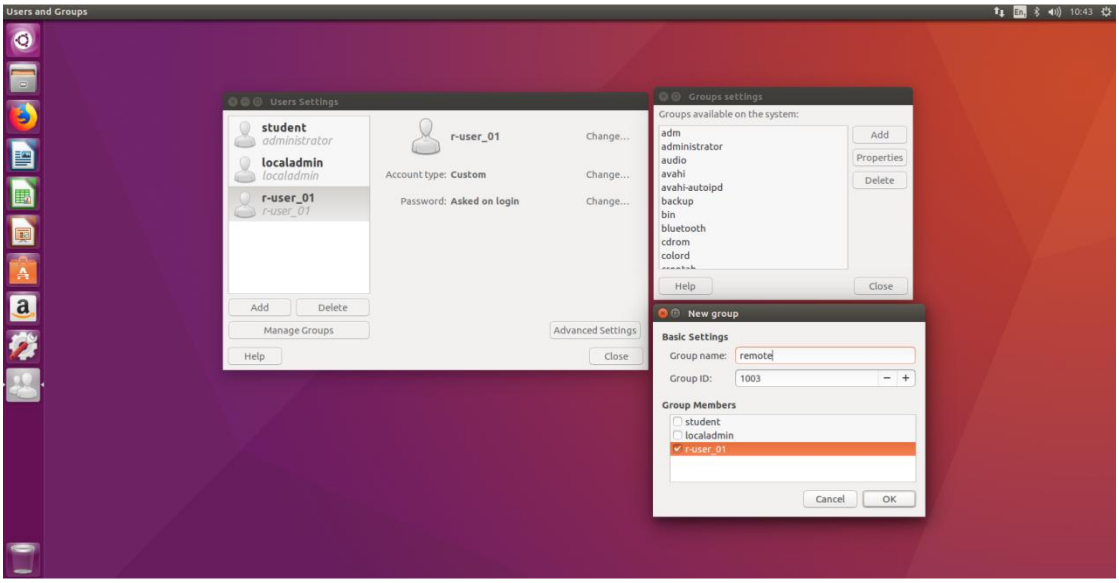
Password: Asked on login

Name: r-user_01

Username: r-user_01

Username must consist of:
- lower case letters from the English alphabet
- digits
- any of the characters ".", "-" and "_"

In order to create a new user account called r-user_01 that belongs to the remote group, the following command must be executed first: “sudo apt install gnome-system-tools”. The command “sudo apt install gnome-system-tools” will install the application called “Users and Groups”, this application will allow the creation and management of users and groups. In order to create a new user the application “Users and Groups” should be launched and then the button called “Add” should be pressed, once the button is pressed the user details should be provided, in the image from above the details provided are: “r-user_01” for both name and username of the user.



Users Settings

student
administrator

localadmin
localadmin

r-user_01
r-user_01

r-user_01

Account type: Custom

Password: Asked on login

Groups settings

Groups available on the system:

adm
administrator
audio
avahi
avahi-autoipd
backup
bin
bluetooth
cdrom
colord

New group

Basic Settings

Group name: remote

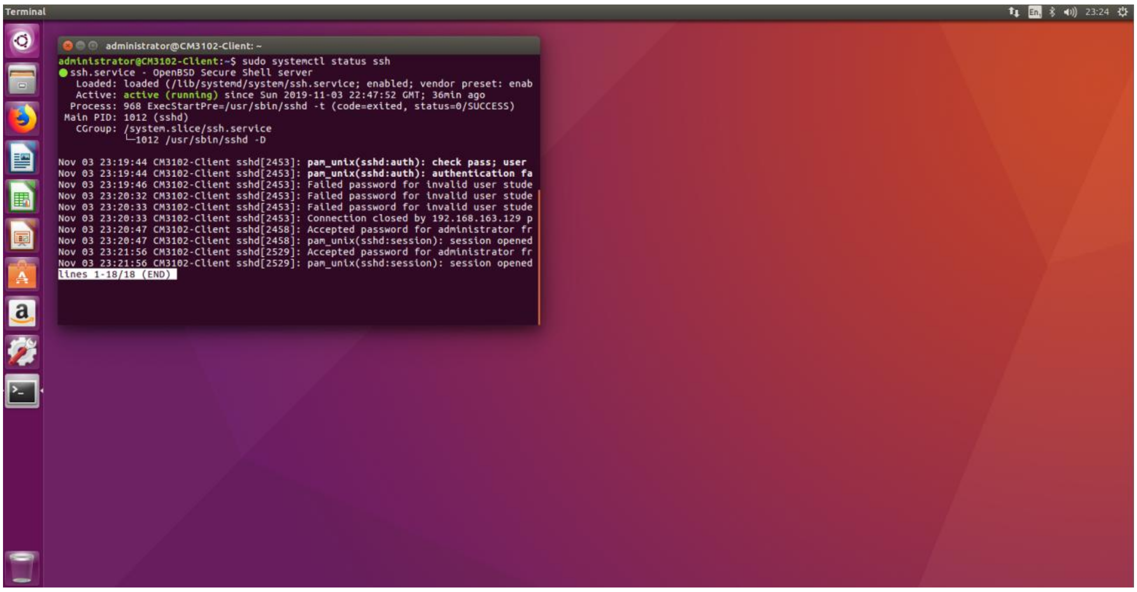
Group ID: 1003

Group Members

□ student
□ localadmin
☑ r-user_01

After the name and username of the user have been provided, the user will have to be provided a password, this step is not present in the previous image. In order to create a new group called remote and add the new user called “r-user_01” to the remote group the following steps have to be performed: press the button called “Manage Groups”, press the button “Add” from the groups settings window, provide the group name, which is in our case “remote” (the upper case version of the word has been tried but an invalid input message has been given by the gnome-system-tools, the image above presents the lower case version of the word), select the user that will be part of the group and click the button called “ok”.

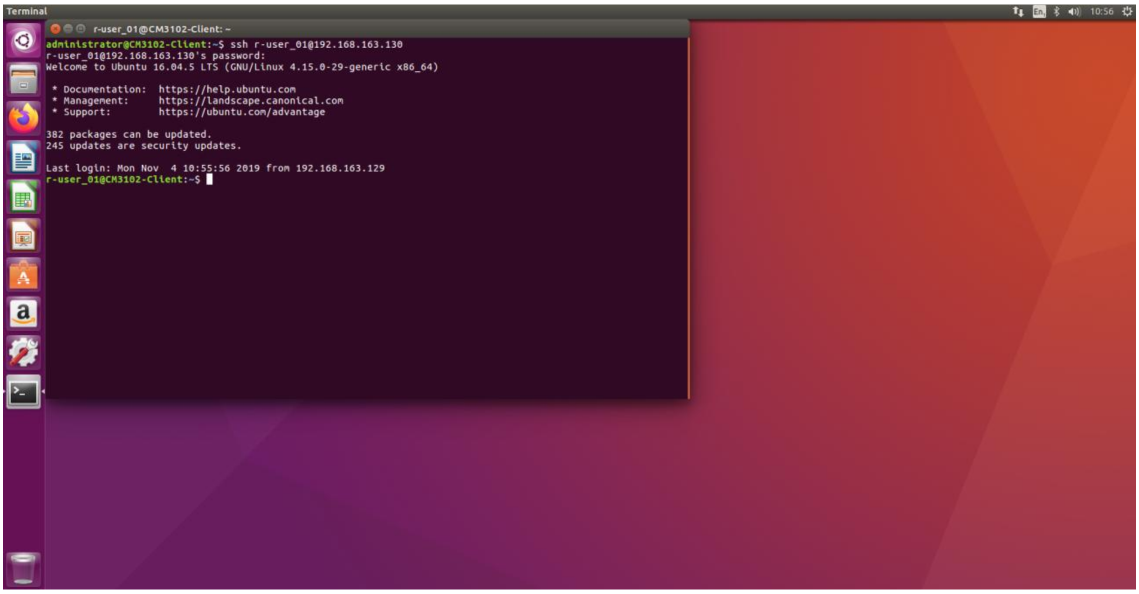
2.4. Remotely login to the CM3102_Serve_VM via SSH using newly created user account r-user_01. (N.B. You might need to start an SSH daemon on the server (/etc/init.d/sshd) to make this work.)



administrator@CM3102-Client: ~

```
administrator@CM3102-Client:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
   Active: active (running) since Sun 2019-11-03 22:47:52 GMT; 36min ago
  Process: 968 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1012 (sshd)
   CGroup: /system.slice/ssh.service
           └─1012 /usr/sbin/sshd -D

Nov 03 23:19:44 CM3102-Client sshd[2453]: pam_unix(sshd:auth): check pass; user
Nov 03 23:19:44 CM3102-Client sshd[2453]: pam_unix(sshd:auth): authentication fa
Nov 03 23:19:46 CM3102-Client sshd[2453]: Failed password for invalid user stude
Nov 03 23:20:32 CM3102-Client sshd[2453]: Failed password for invalid user stude
Nov 03 23:20:33 CM3102-Client sshd[2453]: Failed password for invalid user stude
Nov 03 23:20:33 CM3102-Client sshd[2453]: Connection closed by 192.168.163.129 p
Nov 03 23:20:47 CM3102-Client sshd[2458]: Accepted password for administrator fr
Nov 03 23:20:47 CM3102-Client sshd[2458]: pam_unix(sshd:session): session opened
Nov 03 23:21:56 CM3102-Client sshd[2529]: Accepted password for administrator fr
Nov 03 23:21:56 CM3102-Client sshd[2529]: pam_unix(sshd:session): session opened
lines 1-18/18 (END)
```

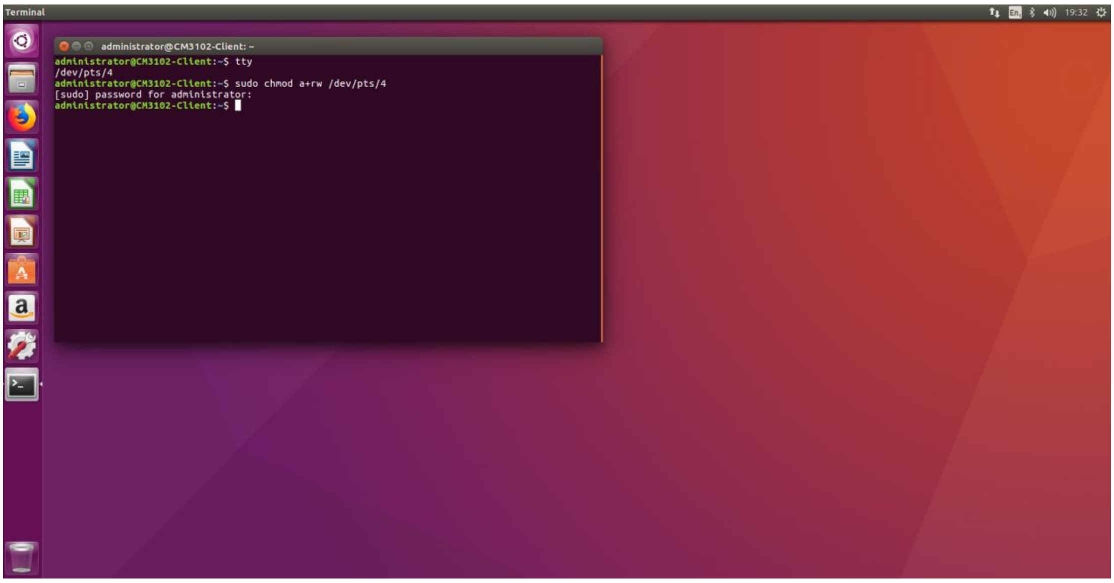
In order to login to the CM3102_Serve_VM via SSH using newly created user account r-user_01, the secure shell server must be enabled in order to establish a connection between the client virtual machine and the server virtual machine. In order to verify if the secure shell server is enabled on the server virtual machine the following command must be executed: “sudo systemctl status ssh”. The command “sudo systemctl status ssh” will show the current status of the secure shell server, as we can see in the image above, the secure shell server is enabled and active, which will allow the client virtual machine to establish a connection with the server virtual machine.



```
administrator@CM3102-Client:~$ ssh r-user_01@192.168.163.130
r-user_01@192.168.163.130's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

382 packages can be updated.
245 updates are security updates.

Last login: Mon Nov  4 10:55:56 2019 from 192.168.163.129
r-user_01@CM3102-Client:~$ 
```

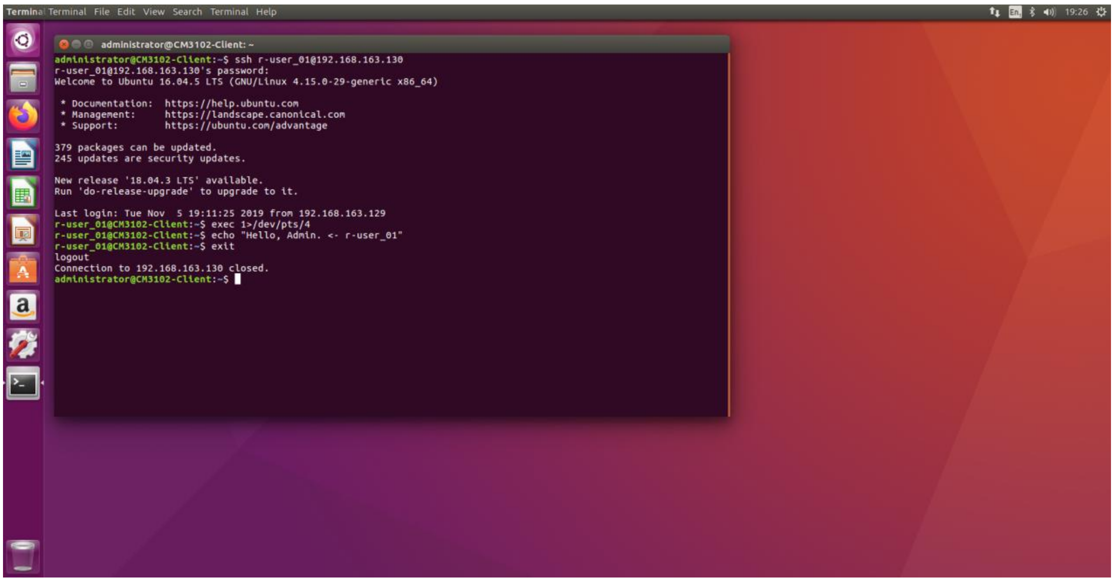
In order to establish a connection between the client virtual machine and the server virtual machine after the secure shell server has been verified on the server virtual machine, the following command must be executed on the client virtual machine: “ssh r-user_01@192.168.163.130”. The command “ssh r-user_01@192.168.163.130” will allow the connection to the server virtual machine using the remote user called “r-user_01”, once this command has been executed the password of the “r-user_01” has to be provided in order to establish the connection. As we can see in the image above the connection has been established successfully after the correct password has been provided, the current user now is “r-user_01”.

2.5. Remotely run a shell script or command on the server that send a message of your choice to the admin user on the CM3102_Server_VM, making sure that the remote users are connected to the server until they want to exit.



```
administrator@CM3102-Client:~$ tty
/dev/pts/4
administrator@CM3102-Client:~$ sudo chmod a+rw /dev/pts/4
[sudo] password for administrator:
administrator@CM3102-Client:~$ 
```

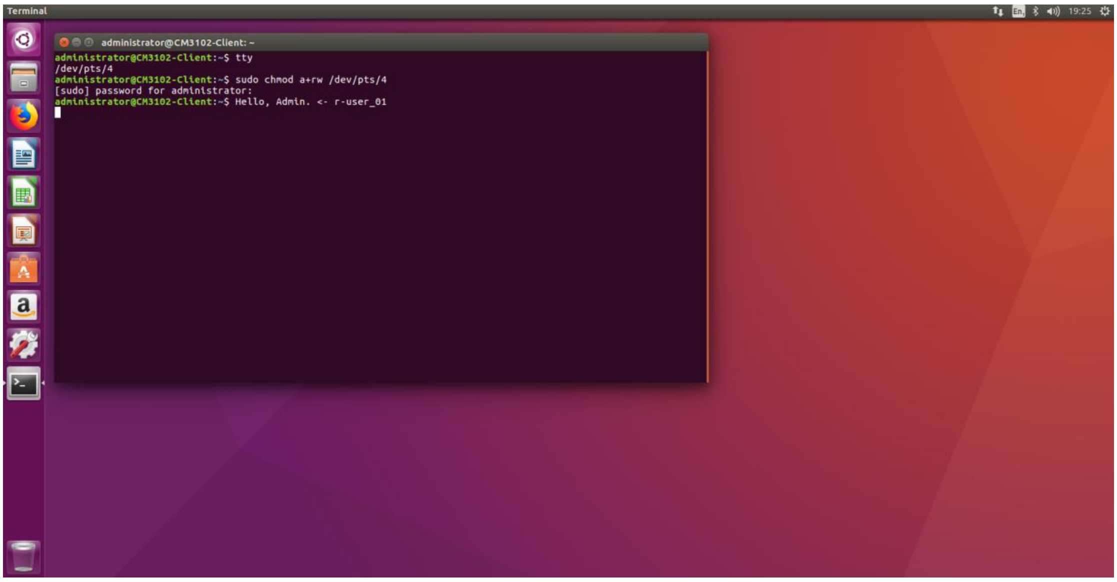
In order to run a shell script or command on the server that send a message to the administrator user on the CM3102_Serve_VM, making sure that the remote users are connected to the server until they want to exit, the following commands must be executed on the server virtual machine: “tty” and “sudo chmod a+rw /dev/pts/4”. The command “tty” will show the terminal that the user called administrator is using, in our case we have the result “/dev/pts/4”, this terminal will be used in order to send the message from the client virtual machine but to do so, we need to give access to all users to read and write to the file, this is done using the command “sudo chmod a+rw /dev/pts/4”, once this is done the client virtual machine will be able to send messages to the administrator user on the server virtual machine.



Terminal Terminal File Edit View Search Terminal Help
administrator@CM3102-Client: ~

```
administrator@CM3102-Client:~$ ssh r-user_01@192.168.163.130
r-user_01@192.168.163.130's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

379 packages can be updated.
245 updates are security updates.

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Nov  5 19:11:25 2019 from 192.168.163.129
r-user_01@CM3102-Client:~$ exec 1>/dev/pts/4
r-user_01@CM3102-Client:~$ echo "Hello, Admin. <- r-user_01"
r-user_01@CM3102-Client:~$ exit
logout
Connection to 192.168.163.130 closed.
administrator@CM3102-Client:~$ 
```

In order to establish a connection to the server virtual machine from the client virtual machine using secure shell and the remote user called “r-user_01”, the following commands must be executed: “ssh r-user_01@192.168.163.130”, “exec 1>/dev/pts/4”, “echo “Hello, Admin. <- r-user_01” and “exit”. The command “ssh r-user_01@192.168.163.130” will establish the connection between the client virtual machine and the server virtual machine once the password of the r-user_01 is provided, the command exec 1>/dev/pts/4 will allow the commands wrote by the r-user_01 to be executed on the administrator user terminal present in the server virtual machine, the command “echo ‘Hello, Admin. <- r-user_01’” will print a message on the administrator’s user terminal and the command “exit” will close the connection to the server virtual machine.



```
administrator@CM3102-Client:~$ tty
/dev/pts/4
administrator@CM3102-Client:~$ sudo chmod a+rw /dev/pts/4
[sudo] password for administrator:
administrator@CM3102-Client:~$ Hello, Admin. <- r-user_01

```

In order to visualise the message send from the client virtual machine using the secure shell and the remote user r-user_01, the administrator’s user terminal on the server virtual machine has to be checked. As we can see in the image from above the message send from the client virtual machine using the secure shell, the remote user r-user_01, the host 192.168.163.130 and the command “sudo echo ‘Hello, Admin. <- r-user_01’ ” was completed successfully, the message was received by the administrator user on the server virtual machine and can be visualised. The client will be able to run scripts or commands on the server virtual machine until the command “exit” will be executed which will close the connection between the client and server.