College of Computer Training (CCT)

Module Title:	Operating Systems
Programme Title / Year	BSc 1st year
Lecturer Name:	Michael Weiss, mweiss@cct.ie
Student Names:	Lais Ferreira dos Santos
Student Nos.:	2020311
Assignment Due Date:	8th May 2021 @23.55

Academic Year: **Year 1** ■ ☐ **Year 2** ☐ **Year 3**

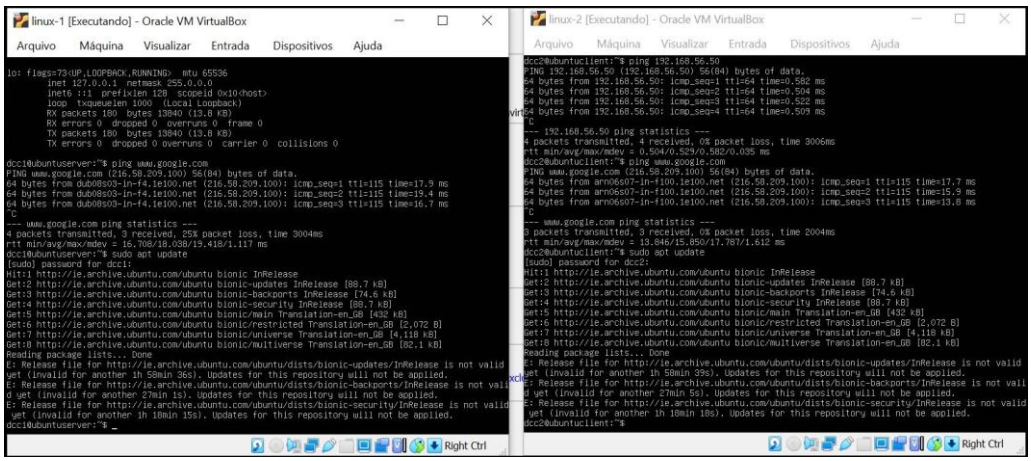
DECLARATION

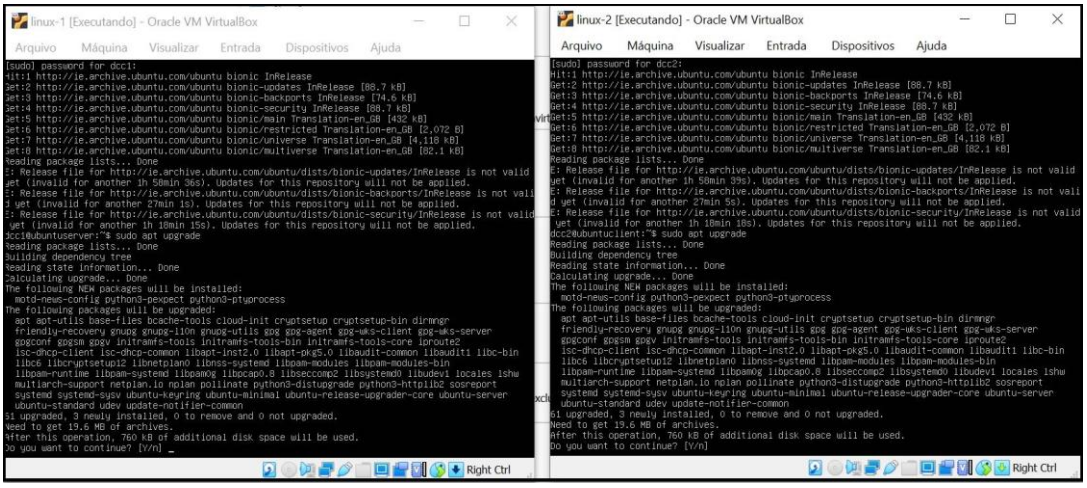
I, Lais Ferreia dos Santos, confirm that by submitting, or causing the attached assignment to be submitted, to CCT, I have not plagiarised any other person's work in this assignment and except where appropriately acknowledged, this assignment is my own work, has been expressed in my own words, and has not previously been submitted for assessment.

Contents

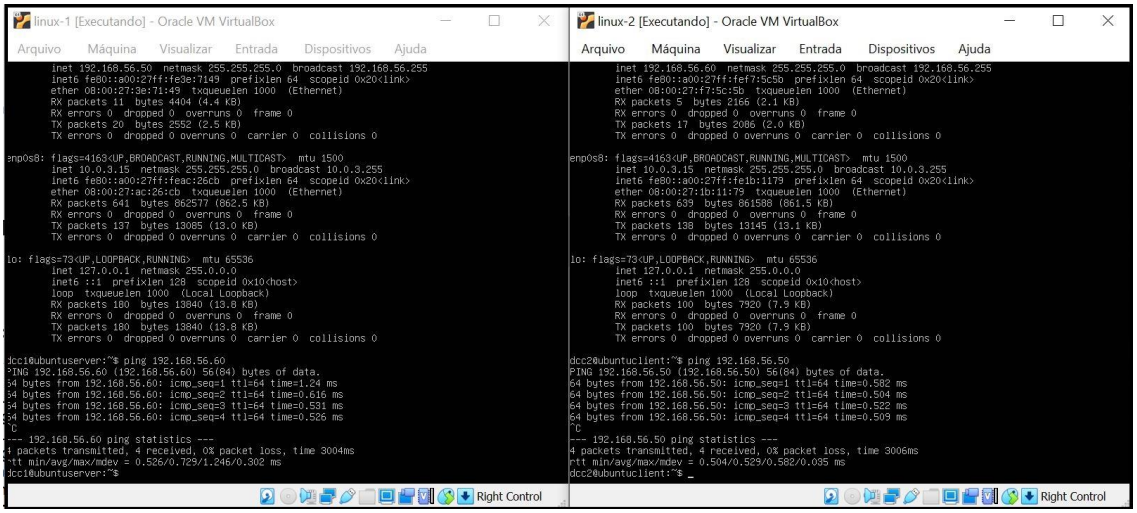
CONSULT & CONNECT LTD

1.Installing Client/Server Web



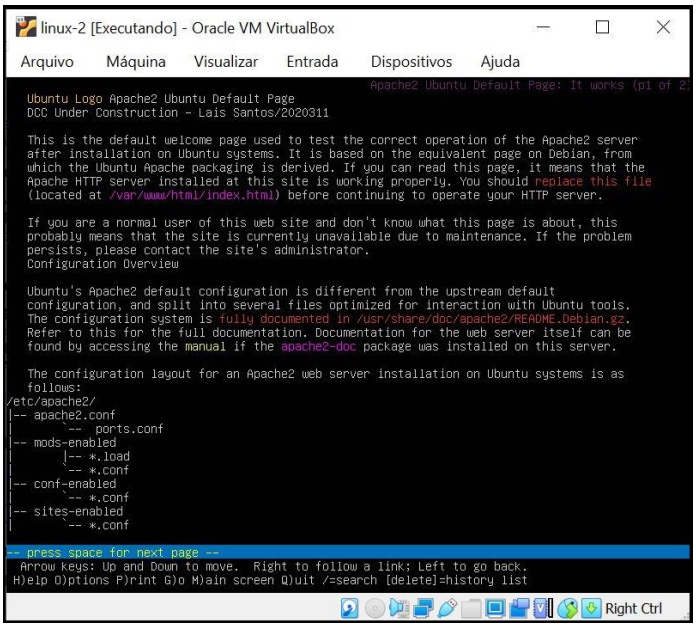


1.1. Ubuntu 18 was installed into two machines, and both were upgraded and updated using the commands **sudo apt upgrade/ sudo apt upgrade,** respectively.



1.2. Both machines are working properly and pinging each other.

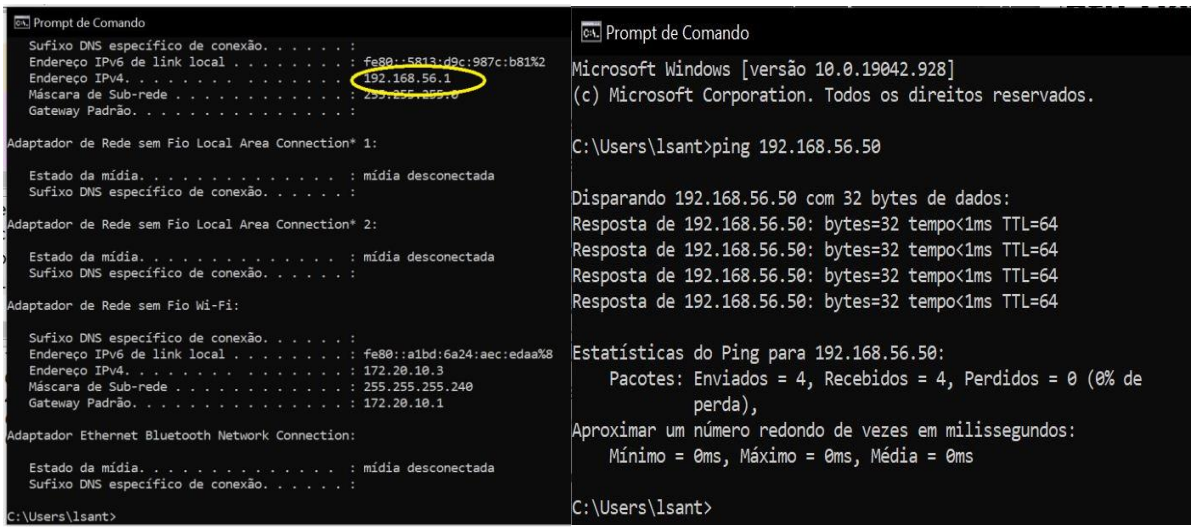
1.2.1. Configurating IP address: **sudo ifconfig enp0s3 (IP ADDRESS) netmask 255.255.255.0**



1.3.A Server machine was created by installing Apache, using the command **sudo apt install apache2.** Using the Client machine, the on the Ubuntu server could be accessed using the follow command **sudo lynx 192.168.56.60.**

1.4. The Apache can be accessed by a web browser.

1.4.1. For editing the Apache web page, the command **sudo nano/var/www/index/html** was used.



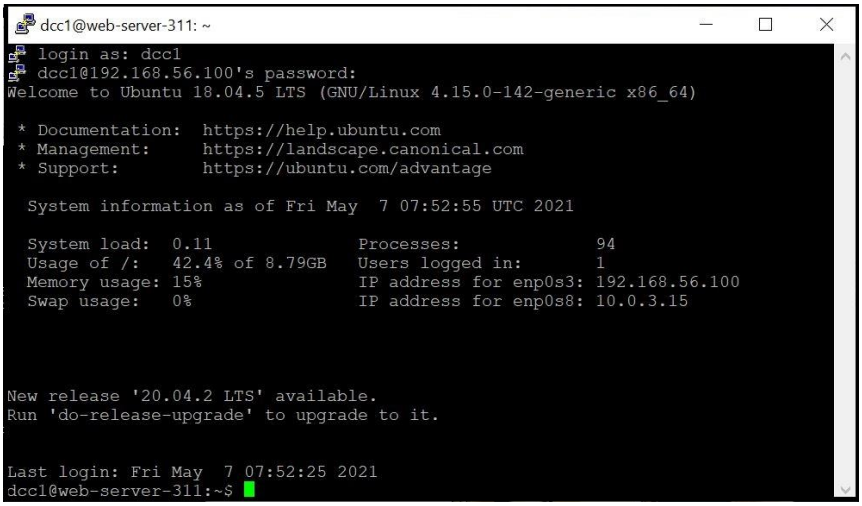
4 0.000662	192.168.56.105	192.168.56.1	ICMP	74	Echo (ping) reply id=0x0001, seq=52/13312, ttl
5 1.008338	192.168.56.1	192.168.56.105	ICMP	74	Echo (ping) request id=0x0001, seq=53/13568, ttl
6 1.008696	192.168.56.105	192.168.56.1	ICMP	74	Echo (ping) reply id=0x0001, seq=53/13568, ttl
7 2.015481	192.168.56.1	192.168.56.105	ICMP	74	Echo (ping) request id=0x0001, seq=54/13824, ttl

1.5. From the Host computer I have done the to the Server machine and the ICMP was identified using the Wireshark.

Time	Source	Destination	Protocol	Length	Info
10 0.021121	192.168.56.105	192.168.56.1	TCP	1514	80 → 50678 [ACK] Seq=1461 Ack=589 Win=64128 Len=146…
11 0.021281	192.168.56.1	192.168.56.105	TCP	54	50678 → 80 [ACK] Seq=589 Ack=2921 Win=262656 Len=0
12 0.021555	192.168.56.105	192.168.56.1	HTTP	638	HTTP/1.1 200 OK (text/html)
13 0.062202	192.168.56.1	192.168.56.105	TCP	54	50678 → 80 [ACK] Seq=589 Ack=3505 Win=262144 Len=0
14 5.021297	192.168.56.105	192.168.56.1	TCP	60	80 → 50678 [FIN, ACK] Seq=3505 Ack=589 Win=64128 Le…

1.6. From the Host computer the Server was accessed using the browser and utilizing the Wireshark we can see the HTTP contents being transferred from the Server to the Client.

2. Installing The Open Secure Shell (SSH)

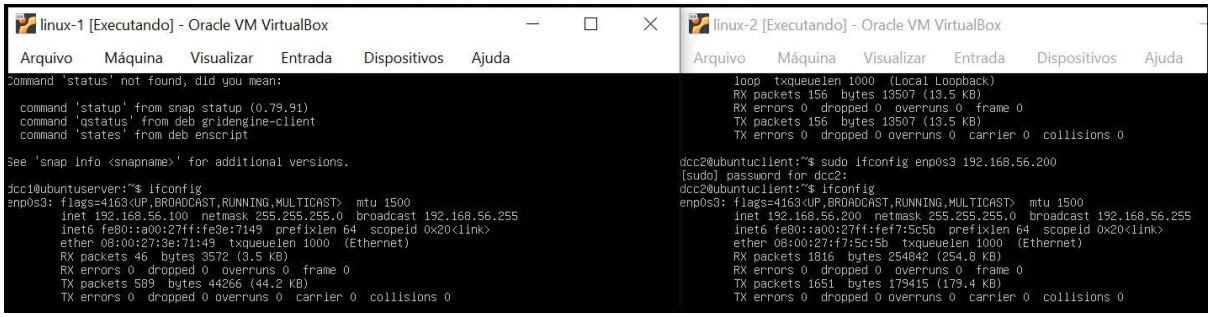


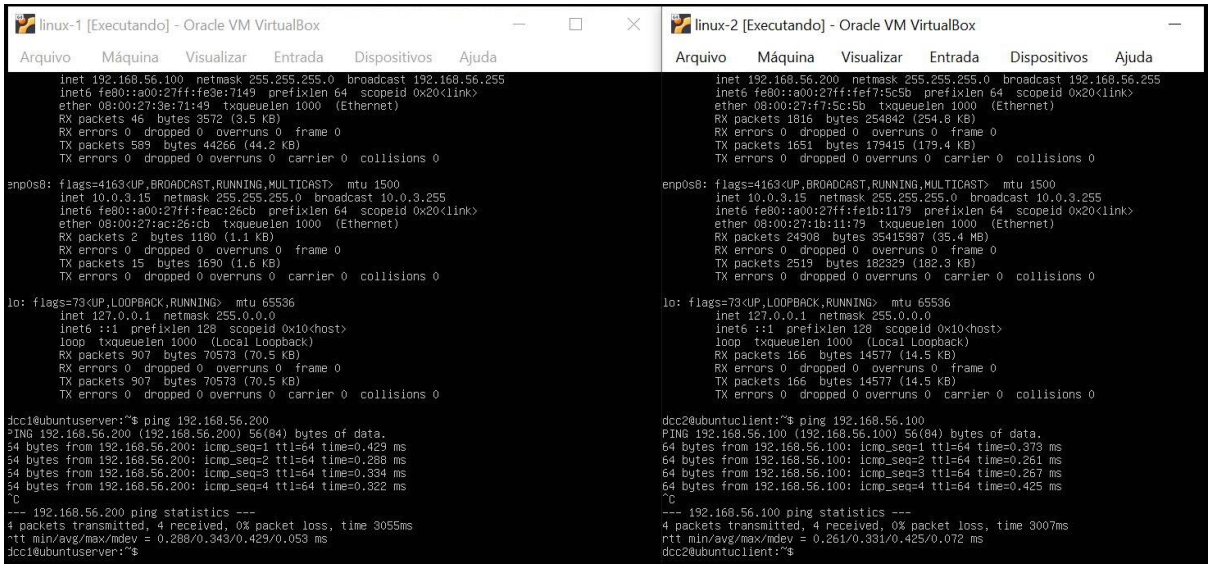
2.1. The Open Secure Shell needs to be installed in the Server machine (sudo apt install openssh-server), enable the port 22 (sudo nano /etc/ssh/sshd_config). Host operating system logging into the server using SSH.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.105	192.168.56.1	SSH	182	Server: Encrypted packet (len=128)
2	0.041429	192.168.56.1	192.168.56.105	TCP	54	50354 → 22 [ACK] Seq=1 Ack=129 Win=8210 Len=0
3	1.023880	192.168.56.105	192.168.56.1	SSH	182	Server: Encrypted packet (len=128)
4	1.065252	192.168.56.1	192.168.56.105	TCP	54	50354 → 22 [ACK] Seq=1 Ack=257 Win=8210 Len=0
5	2.047876	192.168.56.105	192.168.56.1	SSH	182	Server: Encrypted packet (len=128)
6	2.088188	192.168.56.1	192.168.56.105	TCP	54	50354 → 22 [ACK] Seq=1 Ack=385 Win=8209 Len=0
7	3.071810	192.168.56.105	192.168.56.1	SSH	182	Server: Encrypted packet (len=128)
8	3.112588	192.168.56.1	192.168.56.105	TCP	54	50354 → 22 [ACK] Seq=1 Ack=513 Win=8209 Len=0
9	4.096267	192.168.56.105	192.168.56.1	SSH	182	Server: Encrypted packet (len=128)
10	4.096709	192.168.56.1	192.168.56.105	SSH	214	Client: Encrypted packet (len=160)
11	4.096896	192.168.56.105	192.168.56.1	SSH	102	Server: Encrypted packet (len=48)
12	4.136904	192.168.56.1	192.168.56.105	TCP	54	50354 → 22 [ACK] Seq=161 Ack=689 Win=8208 Len=0
13	5.120310	192.168.56.105	192.168.56.1	SSH	182	Server: Encrypted packet (len=128)
14	5.161044	192.168.56.1	192.168.56.105	TCP	54	50354 → 22 [ACK] Seq=161 Ack=817 Win=8207 Len=0
15	6.144601	192.168.56.105	192.168.56.1	SSH	182	Server: Encrypted packet (len=128)
16	6.184570	192.168.56.1	192.168.56.105	TCP	54	50354 → 22 [ACK] Seq=161 Ack=945 Win=8207 Len=0
17	7.168311	192.168.56.105	192.168.56.1	SSH	182	Server: Encrypted packet (len=128)
18	7.207865	192.168.56.1	192.168.56.105	TCP	54	50354 → 22 [ACK] Seq=161 Ack=1073 Win=8206 Len=0

2.2. Communication encrypted between the Client system and ubuntuserver.

3. IP Address configuration and Hostname management





3.1. Configurating IP Address

3.1.1. cd /etc/netplan

Open a new file **(sudo nano 01-config-net.yaml)**

Type: network:

version: 2

renderer: networkd

ethernets:

enp0s3:

dhcp4: false

dhcp6: false

addresses: [192.168.56.100/24]

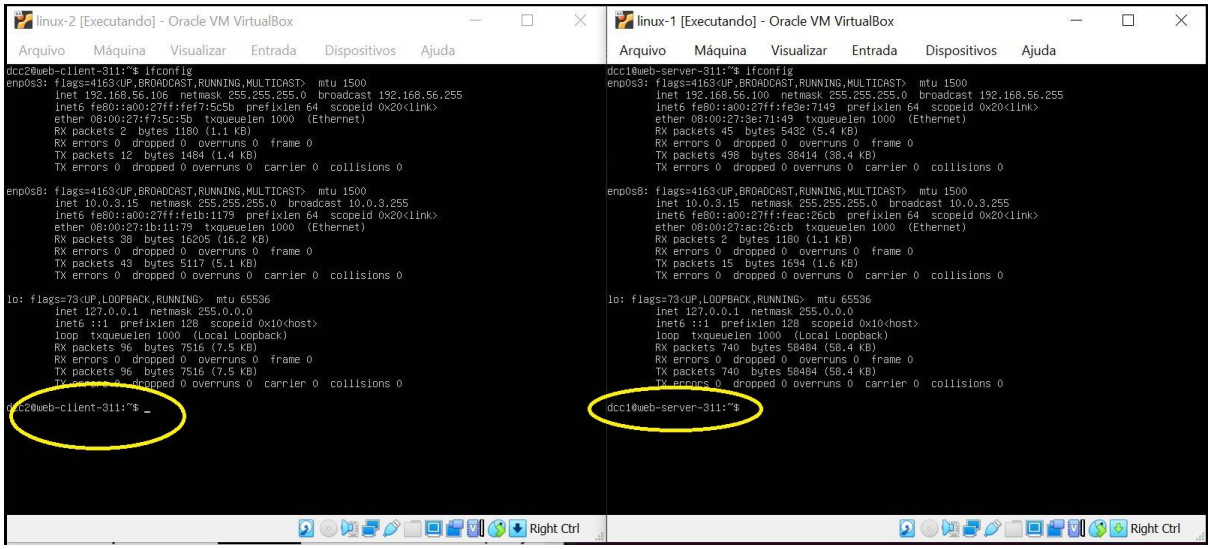
gateway4: 192.168.56.1

nameservers:

addresses: [192.168.56.1]

3.1.2. Activating the configuration

Sudo netplan apply



3.2. Renaming the Host

sudo vim /etc/hostname.
(press I to edit) type the new name.
sudo vim /etc/hosts.
Change the host name.
Reboot.

4. Firewall

```
To                         Action      From
--                         ------      ----
22/tcp                     DENY        Anywhere
80/tcp                     DENY        Anywhere
22/tcp (v6)                DENY        Anywhere (v6)
80/tcp (v6)                DENY        Anywhere (v6)

80/tcp                     DENY OUT    Anywhere
22/tcp                     DENY OUT    Anywhere
80/tcp (v6)                DENY OUT    Anywhere (v6)
22/tcp (v6)                DENY OUT    Anywhere (v6)
```

4.1. SSH and HTTP traffic not being allowed:

For denying this traffics was used the commands:
sudo ufw deny in ssh
sudo ufw deny out ssh
sudo ufw deny in http
sudo ufw deny out http

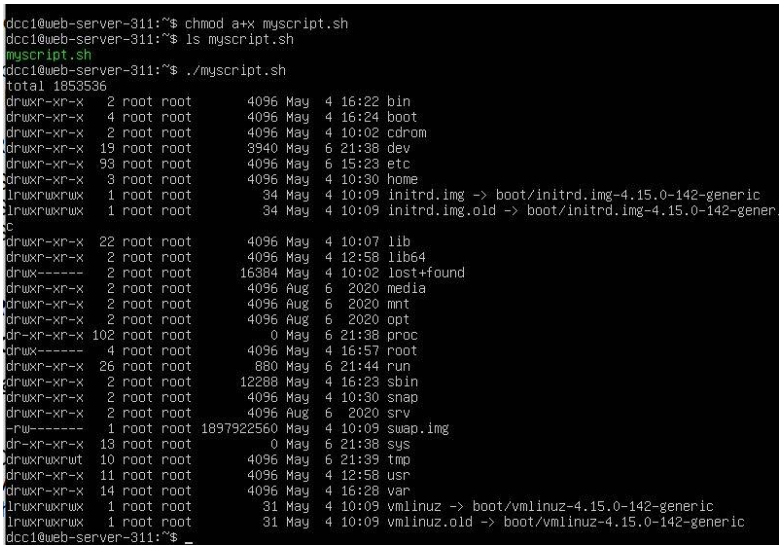
```
To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     DENY        Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                DENY        Anywhere (v6)

80/tcp                     DENY OUT    Anywhere
22/tcp                     ALLOW OUT   Anywhere
80/tcp (v6)                DENY OUT    Anywhere (v6)
22/tcp (v6)                ALLOW OUT   Anywhere (v6)
```

4.2. SSH traffic being allowed:

sudo ufw allow in ssh
sudo ufw allow out ssh

5. Research



5.1. Creating a Simple/Sample Linux Shell? Bash Script

A Shell script is an automated series of Linux commands stored for repeated Later use. The example above shows how it is after the step by step is done.

5.1.1. Steps:

Choose a text editor (vim)
Type **vim myscript.sh** to open the file
Type **i** to open the insertion
Type the commands:
cd /
ls -l and save it.
Type **chmod a+x myscript.sh** to give the permission
The file can be accessed using the command **./myscript.sh**