

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL
PERÚ**

Facultad de Ciencias e Ingeniería

ADMINISTRACIÓN DE SISTEMAS OPERATIVOS

LABORATORIO 3

Alumno: Saymon Nicho

Profesor: Mario Carpio

Horario: 0781

Lima, 24 de septiembre del 2024

Tabla de Contenidos

PREGUNTA 1	4
Figura 1.1. Descarga del instalador de Webex en instancia Windows	4
Figura 1.2. Creación del script de instalación	4
Figura 1.3. Configuración de la tarea a programar	5
Figura 1.4. Programación de la hora de instalación	5
Figura 1.5. Configuración de la acción a realizar con el script	6
Figura 1.6. Resultado de la instalación a la hora programada	6
Figura 1.7. Evidencia de que el programa fue instalado	7
PREGUNTA 2	8
Figura 2.1. Conexión a la instancia Linux	8
Figura 2.2. Revisión del manual del comando crontab	8
Figura 2.3. Creacion del archivo crontab.jfsm0g	9
Figura 2.4. Script que instala nmon y elimina la tarea programada	9
Figura 2.5. Programación de la tarea con el script y el comando crontab -e	10
Figura 2.6. Estado del sistema tras la ejecución de la tarea programada	10
Figura 2.7. Archivo con el resultado de la instalación de nmon	11
Figura 2.8. Programa nmon instalado	11
PREGUNTA 3	12
Figura 3.1. Revisión del manual de nmon	12
Figura 3.2. Uso de nmon para monitorear las cargas de trabajo	12
Figura 3.3. Asesinato del proceso 1546 con kill -9	13
Figura 3.4. Recuperación del archivo desde mi dispositivo local	13
Figura 3.5. Inspección del archivo con NMON Visualizer	14
PREGUNTA 4	15
Figura 4.1. Instalación de samba	15
Figura 4.2. Creación del directorio sambashare	15
Figura 4.3. Configuración de parámetros para el directorio sambashare	16
Figura 4.4. Se descomenta el parámetro map to guest	16
Figura 4.5. Reinicio del servicio y configuración de contraseña	17
Figura 4.6. Creación de una regla de entrada para instancia Linux	17
Figura 4.7. Regla de entrada añadida	18
Figura 4.8. Ingreso de credenciales de acceso	18
Figura 4.9. Creacion de archivo en el directorio L03	19
Figura 4.10. Directorio L03 visto desde la instancia Linux	19
PREGUNTA 5	20
Figura 5.1. Creación de los usuarios	20
Figura 5.2. Creación del grupo SoporteLectura	20
Figura 5.3. Creación del grupo SoporteEscritura	21
Figura 5.4. Creación de la carpeta a compartir	21
Figura 5.5. Asignación de permisos al grupo SoporteLectura	22
Figura 5.6. Asignación de permisos al grupo SoporteEscritura	22

Figura 5.7. Creación de una regla de entrada para instancia Windows	23
Figura 5.8. Acceso a la carpeta compartida por el usuario user01	23
Figura 5.9. Impedimento de creación de un archivo para user01	24
Figura 5.10. Permiso de creación de un archivo para user05	24
Figura 5.11. Archivo creado por user05	25
PREGUNTA 6	26
Figura 6.1. Creación del directorio backup a compartir	26
Figura 6.2. Cambio de permisos del directorio /home/ubuntu/backup	26
Figura 6.3. Se comparte el directorio con otros miembros de la subnet privada	27
Figura 6.4. Reinicio del servicio	27
Figura 6.5. Cliente 1 instala nfs-common	28
Figura 6.6. Archivo creado por el cliente 1 en LinuxResp1	29
Figura 6.7. Montaje y creación de archivos por cada cliente	29
Figura 6.8. Acceso a carpetas de respaldo desde el servidor	29

PREGUNTA 1

En primer lugar, se descarga el instalador de Webex.

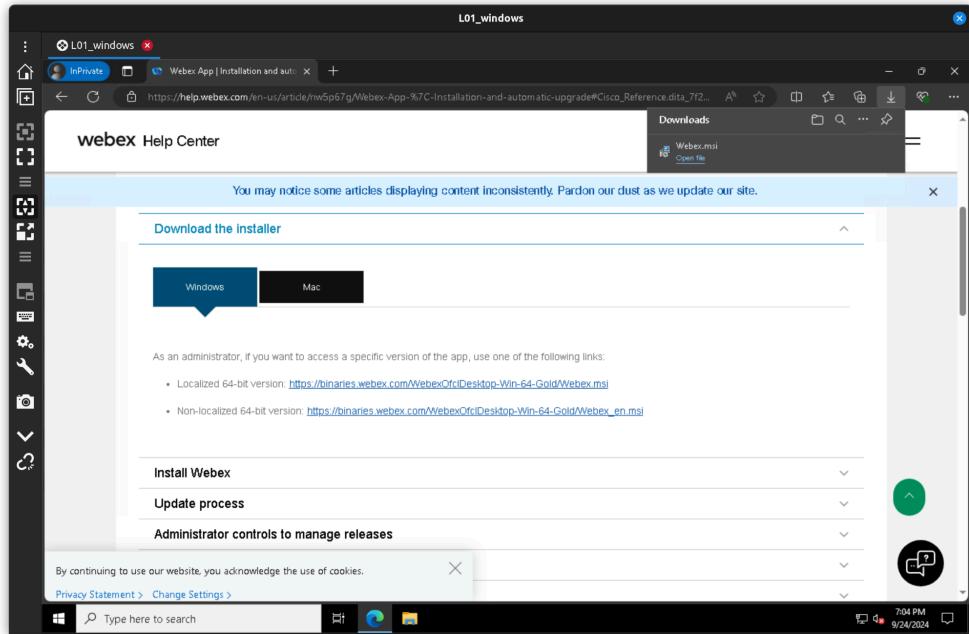


Figura 1.1. Descarga del instalador de Webex en instancia Windows

Luego, se crea el script `sched_webex.bat` el cual contiene las instrucciones para ejecutar el instalador `msiexec`. Se usa la bandera `/qn` para que la instalación sea *quiet* y no interrumpa al usuario. Además, se usa la bandera `/norestart`, la opción de `ACCEPT_EULA` para el acuerdo de términos y condiciones y `ALLUSERS` para que la instalación sea a nivel del equipo.

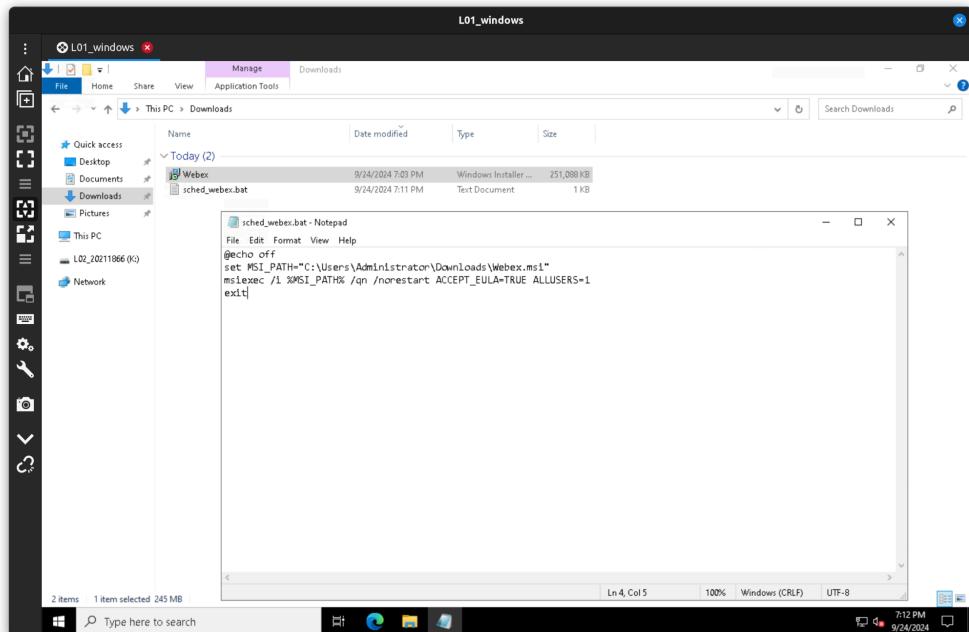


Figura 1.2. Creación del script de instalación

Después, con **Task Scheduler** se programa la instalación con la opción de nueva tarea.

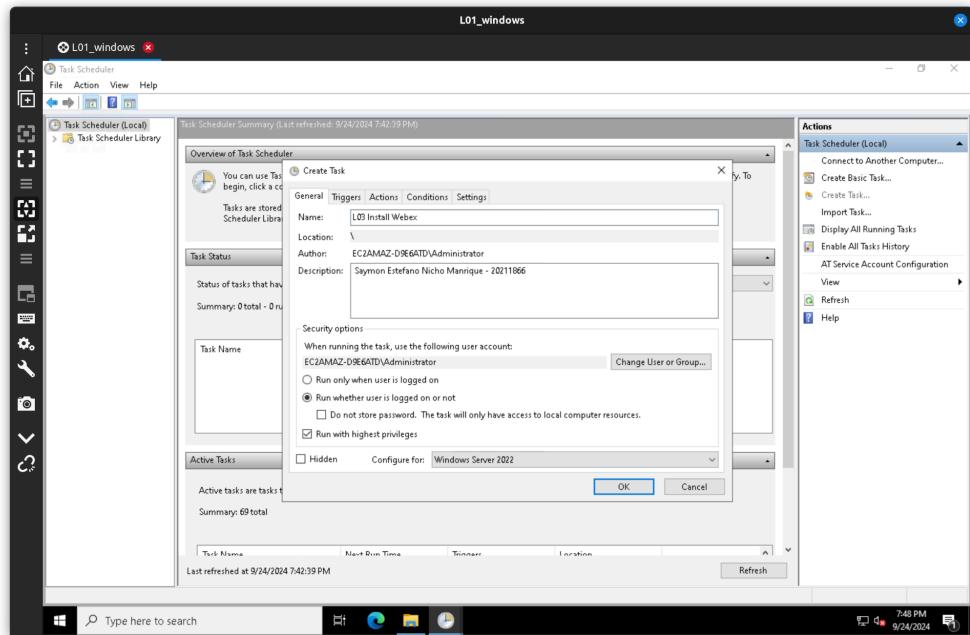


Figura 1.3. Configuración de la tarea a programar

También se configura la hora (a manera de ejemplo se programa dentro de unos 5 minutos).

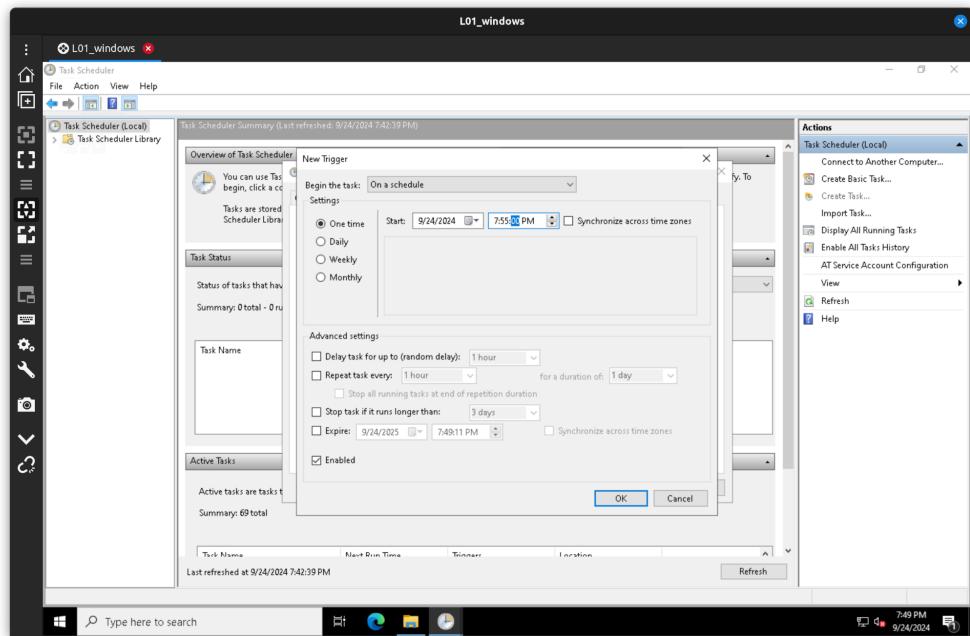


Figura 1.4. Programación de la hora de instalación

Se especifica que la acción a realizar será la ejecución del script de instalación.

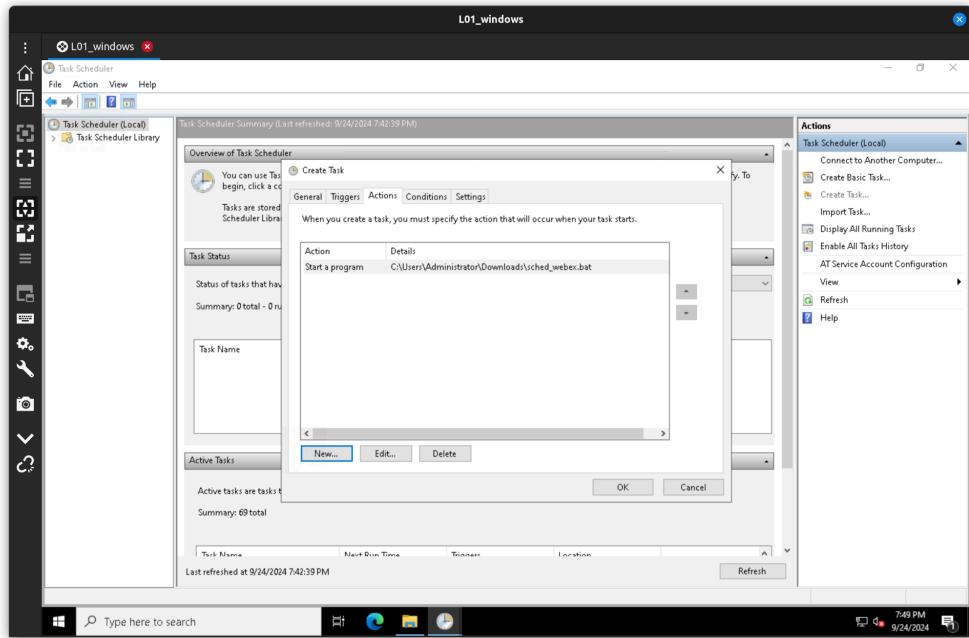


Figura 1.5. Configuración de la acción a realizar con el script

Luego, se simula que el usuario realiza alguna otra tarea y a la hora programada le llega la notificación de que la instalación se dio de forma exitosa.

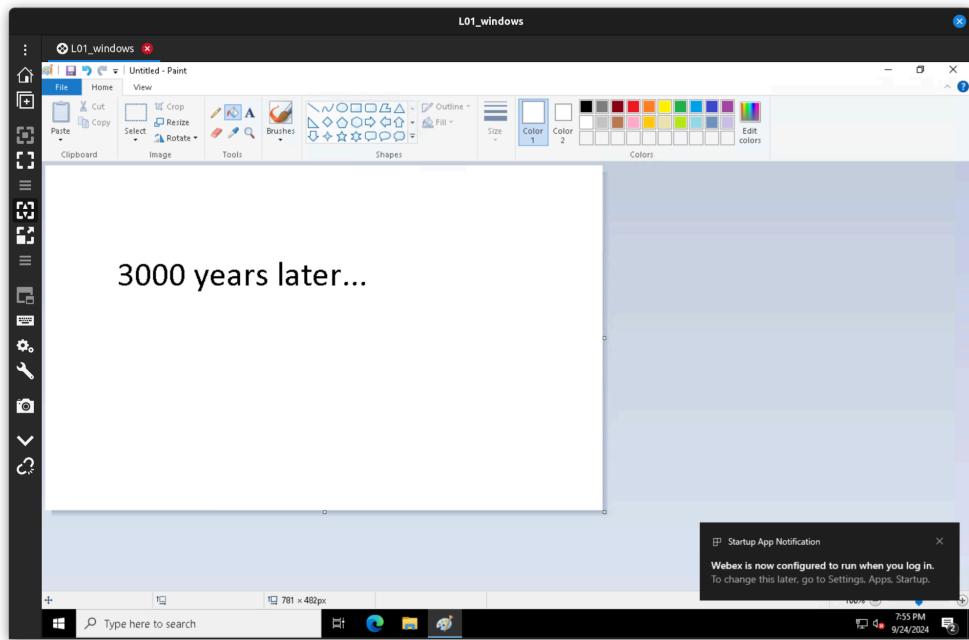


Figura 1.6. Resultado de la instalación a la hora programada

Finalmente, se puede demostrar que la instalación fue exitosa al ubicar los archivos instalados en el directorio de programas del sistema.

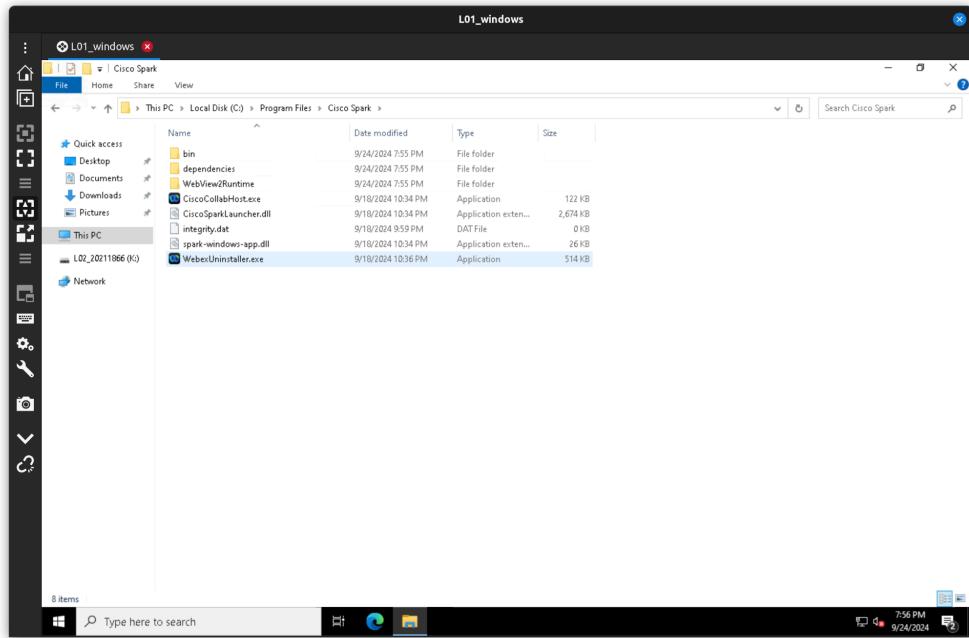
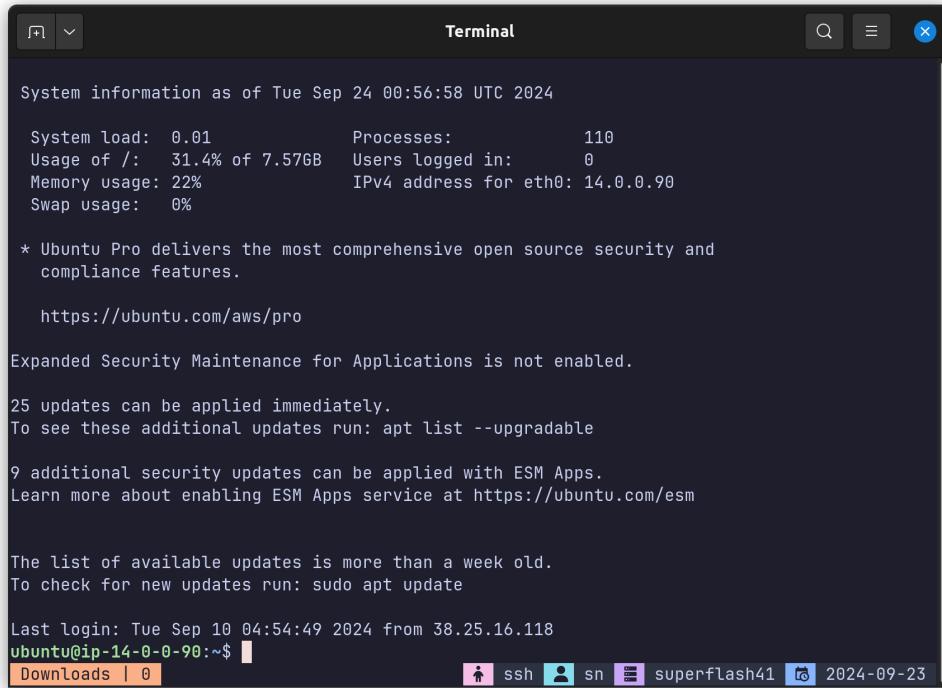


Figura 1.7. Evidencia de que el programa fue instalado

PREGUNTA 2

En primer lugar, se realiza la conexión con la instancia Linux.



```
System information as of Tue Sep 24 00:56:58 UTC 2024

System load: 0.01      Processes: 110
Usage of /: 31.4% of 7.57GB  Users logged in: 0
Memory usage: 22%          IPv4 address for eth0: 14.0.0.90
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

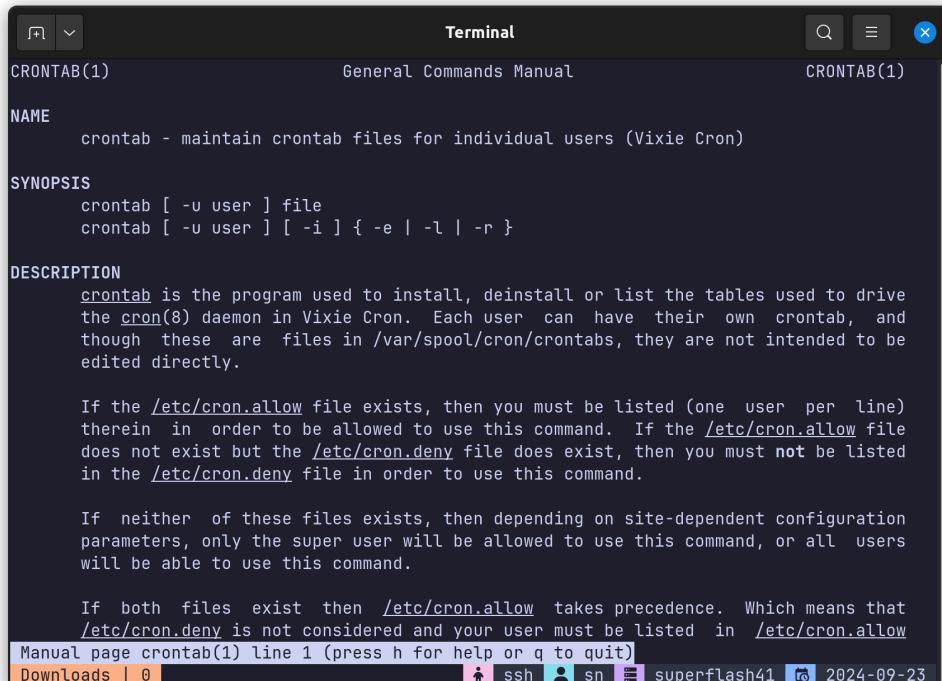
9 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Sep 10 04:54:49 2024 from 38.25.16.118
ubuntu@ip-14-0-0-90:~$ █
Downloads | 0           ssh sn superflash41 2024-09-23
```

Figura 2.1. Conexión a la instancia Linux

Se revisa el manual de **crontab** para obtener mayor información acerca de las banderas y opciones que este comando ofrece.



```
CRONTAB(1)          General Commands Manual          CRONTAB(1)

NAME
    crontab - maintain crontab files for individual users (Vixie Cron)

SYNOPSIS
    crontab [ -u user ] file
    crontab [ -u user ] [ -i ] { -e | -l | -r }

DESCRIPTION
    crontab is the program used to install, deinstall or list the tables used to drive
    the cron(8) daemon in Vixie Cron. Each user can have their own crontab, and
    though these are files in /var/spool/cron/crontabs, they are not intended to be
    edited directly.

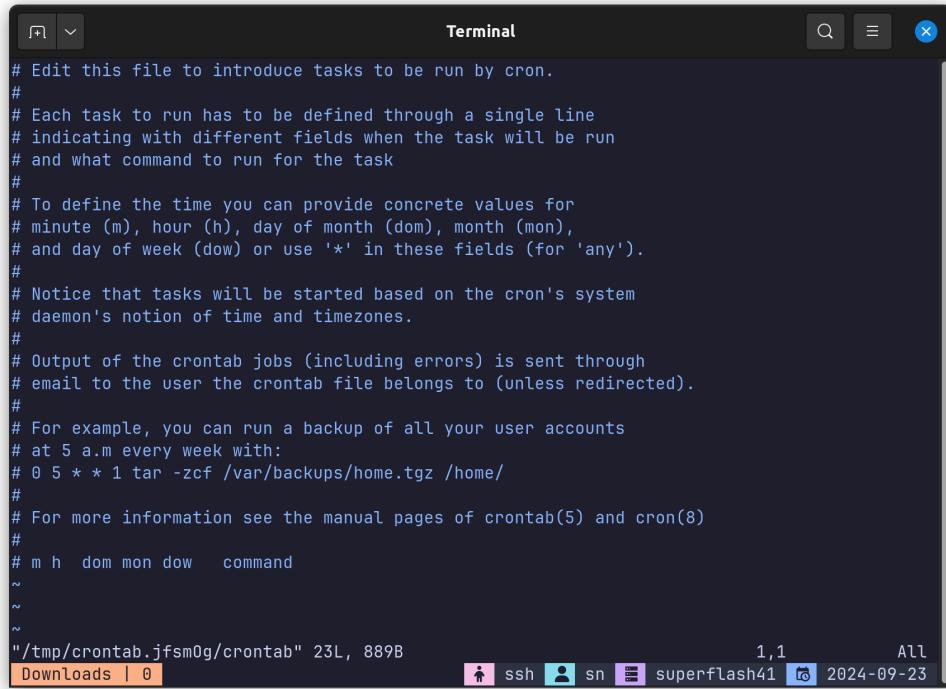
    If the /etc/cron.allow file exists, then you must be listed (one user per line)
    therein in order to be allowed to use this command. If the /etc/cron.allow file
    does not exist but the /etc/cron.deny file does exist, then you must not be listed
    in the /etc/cron.deny file in order to use this command.

    If neither of these files exists, then depending on site-dependent configuration
    parameters, only the super user will be allowed to use this command, or all users
    will be able to use this command.

    If both files exist then /etc/cron.allow takes precedence. Which means that
    /etc/cron.deny is not considered and your user must be listed in /etc/cron.allow
Manual page crontab(1) line 1 (press h for help or q to quit)
Downloads | 0           ssh sn superflash41 2024-09-23
```

Figura 2.2. Revisión del manual del comando **crontab**

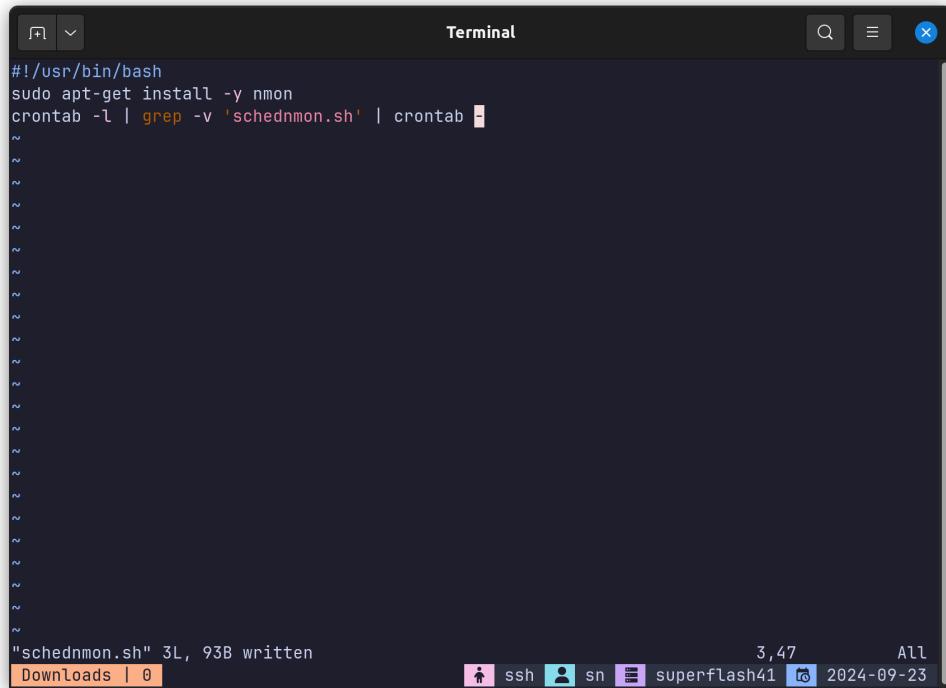
Se usa el comando `crontab -e` para editar el registro de tareas programadas. Dado que es la primera vez que este se ejecuta, se crea el archivo `crontab.jfsm0g`.



```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
~  
~  
~  
~  
~/tmp/crontab.jfsm0g/crontab" 23L, 889B 1,1 All  
Downloads | 0 ssh sn superflash41 2024-09-23
```

Figura 2.3. Creacion del archivo `crontab.jfsm0g`

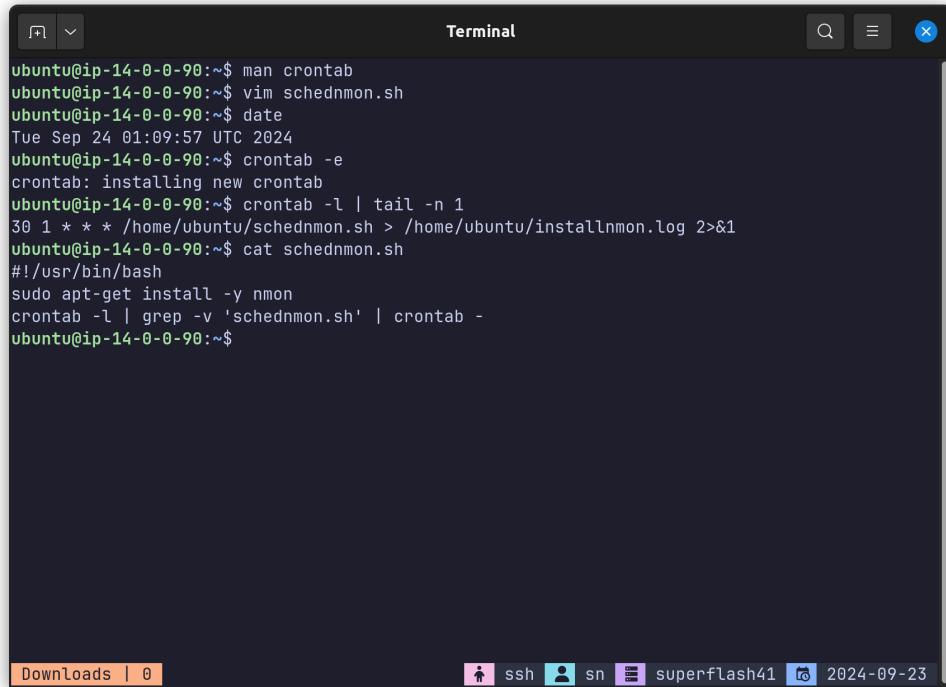
Por ahora no se modifican los contenidos. Se crea primero un script que tenga los comandos de instalación de `nmon` desde el manejador de paquetes de Ubuntu. Asimismo, se incluye una instrucción para eliminar esta tarea de la lista luego de ser ejecutada.



```
#!/usr/bin/bash  
sudo apt-get install -y nmon  
crontab -l | grep -v 'schednmon.sh' | crontab -  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~/tmp/schednmon.sh" 3L, 93B written 3,47 All  
Downloads | 0 ssh sn superflash41 2024-09-23
```

Figura 2.4. Script que instala `nmon` y elimina la tarea programada

Dado que en el sistema la hora actual es de 01:09, se programa la tarea para la 01:30 y se busca que todo el output (ya sea por salida estándar o del log de errores) quede registrado en el archivo `installnmon.log`.

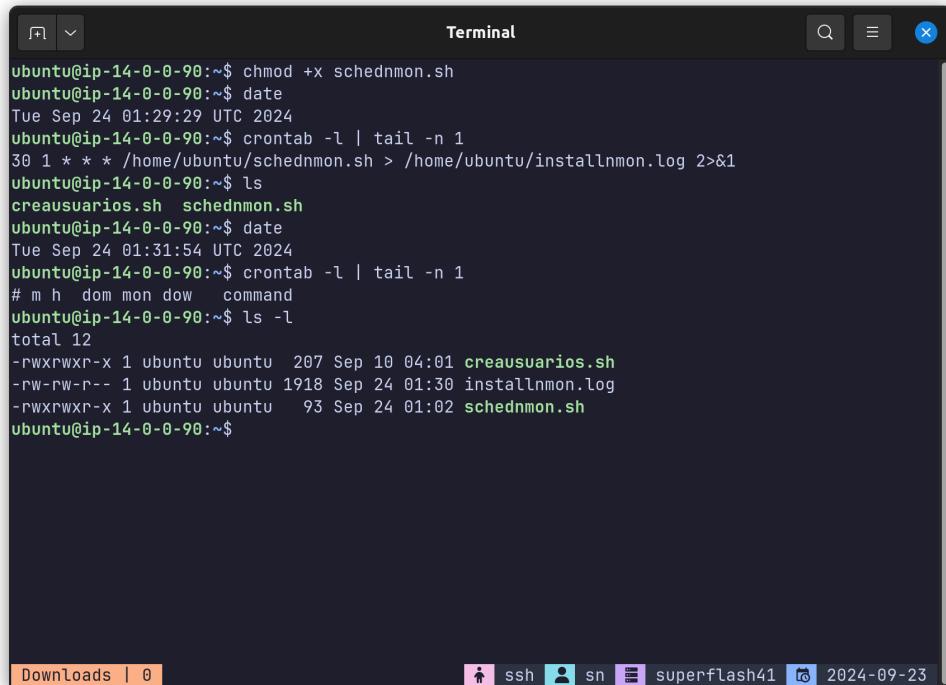


The screenshot shows a terminal window titled "Terminal". The session starts with the user navigating to the man page for crontab, then opening a vim editor to edit the script `schednmon.sh`. The user then runs the command `date` to check the current time (Tue Sep 24 01:09:57 UTC 2024). They proceed to edit the crontab file with `crontab -e`, adding a new entry: `30 1 * * * /home/ubuntu/schednmon.sh > /home/ubuntu/installnmon.log 2>&1`. After saving the changes, they run `cat schednmon.sh` to verify the contents of the script. Finally, they use `sudo apt-get install -y nmon` to install the required package, and then run `crontab -l | grep -v 'schednmon.sh' | crontab -` to apply the changes. The terminal window also displays system status icons at the bottom.

```
ubuntu@ip-14-0-0-90:~$ man crontab
ubuntu@ip-14-0-0-90:~$ vim schednmon.sh
ubuntu@ip-14-0-0-90:~$ date
Tue Sep 24 01:09:57 UTC 2024
ubuntu@ip-14-0-0-90:~$ crontab -e
crontab: installing new crontab
ubuntu@ip-14-0-0-90:~$ crontab -l | tail -n 1
30 1 * * * /home/ubuntu/schednmon.sh > /home/ubuntu/installnmon.log 2>&1
ubuntu@ip-14-0-0-90:~$ cat schednmon.sh
#!/usr/bin/bash
sudo apt-get install -y nmon
crontab -l | grep -v 'schednmon.sh' | crontab -
ubuntu@ip-14-0-0-90:~$
```

Figura 2.5. Programación de la tarea con el script y el comando `crontab -e`

Pasada la hora programada, se nota que el archivo con el log de instalación ya fue generado.

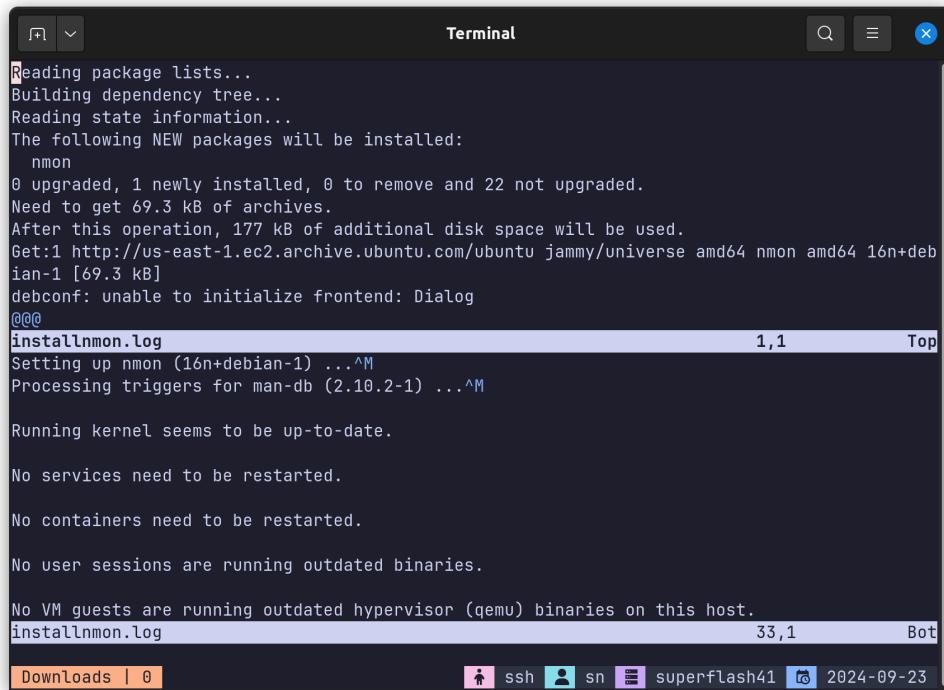


The screenshot shows a terminal window titled "Terminal". The user first checks the date (Tue Sep 24 01:29:29 UTC 2024), then lists the contents of the current directory with `ls`, which shows files `creausuarios.sh` and `schednmon.sh`. The user then runs `crontab -l | tail -n 1` to view the scheduled task, which is the same command as before. Finally, they run `ls -l` to list all files in the directory, showing the newly generated file `installnmon.log` along with the others. The terminal window also displays system status icons at the bottom.

```
ubuntu@ip-14-0-0-90:~$ chmod +x schednmon.sh
ubuntu@ip-14-0-0-90:~$ date
Tue Sep 24 01:29:29 UTC 2024
ubuntu@ip-14-0-0-90:~$ crontab -l | tail -n 1
30 1 * * * /home/ubuntu/schednmon.sh > /home/ubuntu/installnmon.log 2>&1
ubuntu@ip-14-0-0-90:~$ ls
creausuarios.sh  schednmon.sh
ubuntu@ip-14-0-0-90:~$ date
Tue Sep 24 01:31:54 UTC 2024
ubuntu@ip-14-0-0-90:~$ crontab -l | tail -n 1
# m h dom mon dow   command
ubuntu@ip-14-0-0-90:~$ ls -l
total 12
-rwxrwxr-x 1 ubuntu ubuntu  207 Sep 10 04:01 creausuarios.sh
-rw-rw-r-- 1 ubuntu ubuntu 1918 Sep 24 01:30 installnmon.log
-rwxrwxr-x 1 ubuntu ubuntu   93 Sep 24 01:02 schednmon.sh
ubuntu@ip-14-0-0-90:~$
```

Figura 2.6. Estado del sistema tras la ejecución de la tarea programada

Se revisa y se observa como la instalacion se dio de manera exitosa.



```
Reading package lists...
Building dependency tree...
Reading state information...
The following NEW packages will be installed:
  nmon
0 upgraded, 1 newly installed, 0 to remove and 22 not upgraded.
Need to get 69.3 kB of archives.
After this operation, 177 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 nmon amd64 16n+debian-1 [69.3 kB]
debconf: unable to initialize frontend: Dialog
@@@ installnmon.log                                         1,1          Top
Setting up nmon (16n+debian-1) ...^M
Processing triggers for man-db (2.10.2-1) ...^M

Running kernel seems to be up-to-date.

No services need to be restarted.

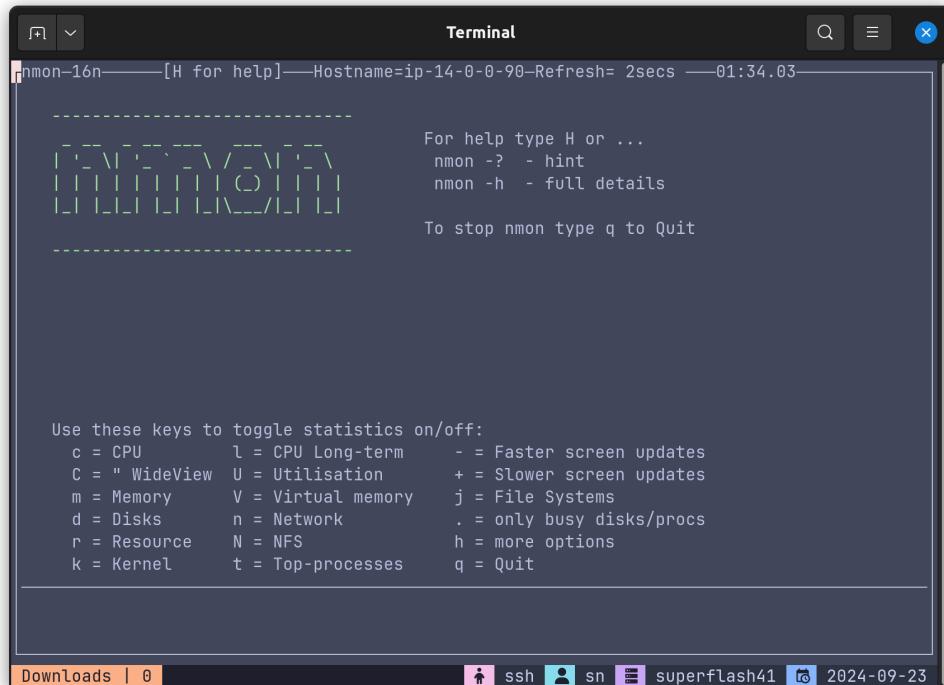
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
installnmon.log                                         33,1          Bot
Downloads | 0                                         ssh sn superflash41 2024-09-23
```

Figura 2.7. Archivo con el resultado de la instalación de **nmon**

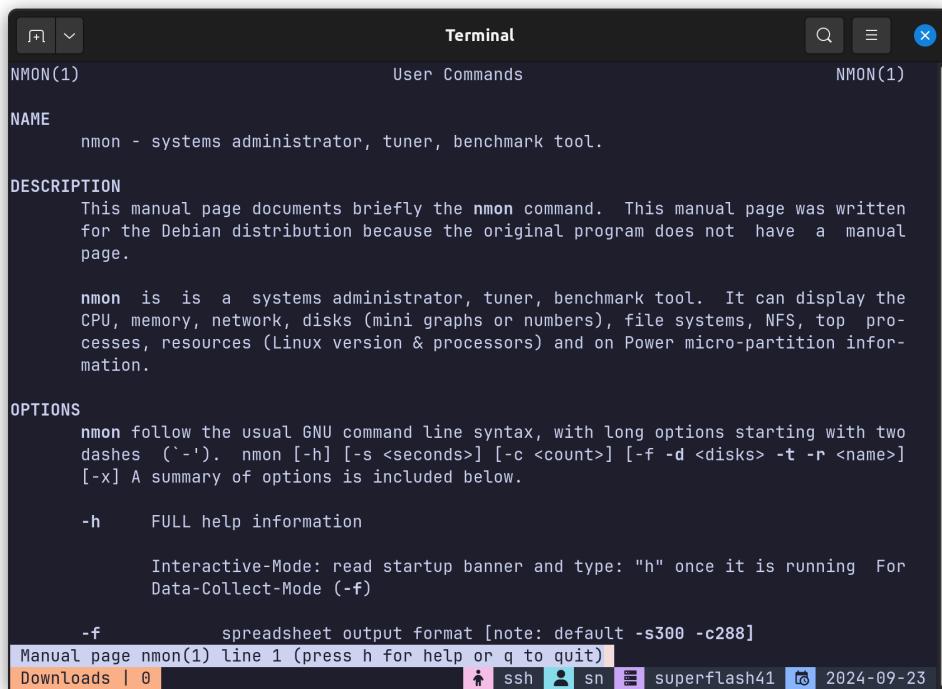
Finalmente, se puede ejecutar **nmon**.



```
nmon-16n———[H for help]——Hostname=ip-14-0-0-90-Refresh= 2secs ——01:34.03-
-----
For help type H or ...
  nmon -?  - hint
  nmon -h  - full details
To stop nmon type q to Quit
-----
Use these keys to toggle statistics on/off:
  c = CPU      l = CPU Long-term      - = Faster screen updates
  C = "WideView"  U = Utilisation      + = Slower screen updates
  m = Memory    V = Virtual memory    j = File Systems
  d = Disks     n = Network           . = only busy disks/procs
  r = Resource   N = NFS              h = more options
  k = Kernel     t = Top-processes    q = Quit
Downloads | 0                                         ssh sn superflash41 2024-09-23
```

Figura 2.8. Programa **nmon** instalado

PREGUNTA 3



The screenshot shows a terminal window with the title "Terminal". The content is the nmon(1) manual page. It includes sections for NAME, DESCRIPTION, and OPTIONS. The DESCRIPTION section explains that nmon is a systems administrator, tuner, benchmark tool. The OPTIONS section details command-line arguments like -h for help, -f for spreadsheet output format, and -s for sampling interval. The bottom of the window shows a status bar with icons for user, ssh, sn, and superflash41, and the date 2024-09-23.

```
NMON(1)                               User Commands                               NMON(1)

NAME
nmon - systems administrator, tuner, benchmark tool.

DESCRIPTION
This manual page documents briefly the nmon command. This manual page was written
for the Debian distribution because the original program does not have a manual
page.

nmon is a systems administrator, tuner, benchmark tool. It can display the
CPU, memory, network, disks (mini graphs or numbers), file systems, NFS, top pro-
cesses, resources (Linux version & processors) and on Power micro-partition infor-
mation.

OPTIONS
nmon follows the usual GNU command line syntax, with long options starting with two
dashes ('-'). nmon [-h] [-s <seconds>] [-c <count>] [-f -d <disks> -t -r <name>]
[-x] A summary of options is included below.

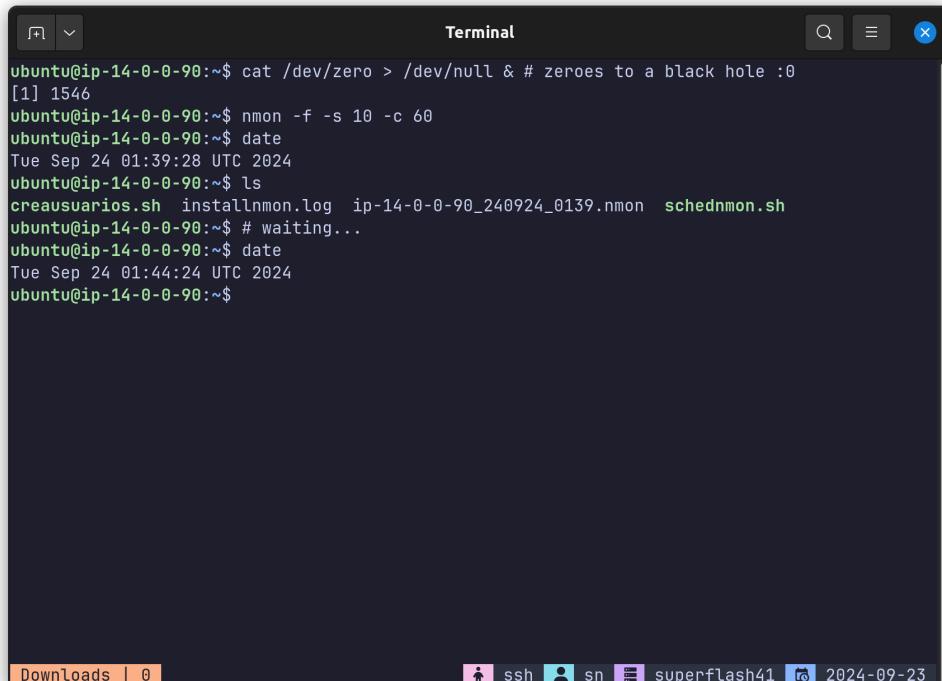
-h      FULL help information

Interactive-Mode: read startup banner and type: "h" once it is running For
Data-Collect-Mode (-f)

-f      spreadsheet output format [note: default -s300 -c288]
Manual page nmon(1) line 1 (press h for help or q to quit)
Downloads | 0   ssh  sn  superflash41  2024-09-23
```

Figura 3.1. Revisión del manual de **nmon**

Como el objetivo es que el monitoreo dure 10 minutos se busca obtener 60 monitoreos instantáneos cada 10 segundos (600 segundos en total).



The screenshot shows a terminal window with the title "Terminal". The user runs the command "cat /dev/zero > /dev/null & # zeroes to a black hole :0" to create a busy loop. Then, they run "nmon -f -s 10 -c 60" to start monitoring. The terminal shows the command being run and the resulting output. The bottom of the window shows a status bar with icons for user, ssh, sn, and superflash41, and the date 2024-09-23.

```
ubuntu@ip-14-0-0-90:~$ cat /dev/zero > /dev/null & # zeroes to a black hole :0
[1] 1546
ubuntu@ip-14-0-0-90:~$ nmon -f -s 10 -c 60
ubuntu@ip-14-0-0-90:~$ date
Tue Sep 24 01:39:28 UTC 2024
ubuntu@ip-14-0-0-90:~$ ls
creausuarios.sh installnmon.log ip-14-0-0-90_240924_0139.nmon schednmon.sh
ubuntu@ip-14-0-0-90:~$ # waiting...
ubuntu@ip-14-0-0-90:~$ date
Tue Sep 24 01:44:24 UTC 2024
ubuntu@ip-14-0-0-90:~$
```

Figura 3.2. Uso de **nmon** para monitorear las cargas de trabajo

Usamos el comando `top` para monitorear el sistema luego de unos 5 minutos. Notamos como el proceso `1546` es el que genera una mayor carga al consumir la mayor cantidad de tiempo del CPU.

```
%Cpu(s): 12.6 us, 87.4 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 949.2 total, 187.2 free, 206.7 used, 555.3 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 580.4 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1546 ubuntu 20 0 6332 2048 1920 R 99.9 0.2 6:17.99 cat
1 root 20 0 101864 12796 8316 S 0.0 1.3 0:01.39 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 slub_flushwq
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
8 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-events+
11 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
12 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthr+
13 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kth+
14 root 20 0 0 0 0 S 0.0 0.0 0:00.07 ksoftirqd/0
15 root 20 0 0 0 0 I 0.0 0.0 0:00.16 rcu_sched
16 root rt 0 0 0 0 S 0.0 0.0 0:00.01 migration/0
17 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0
18 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
19 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs
20 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 inet_frag_wq
22 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kauditd
23 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd

ubuntu@ip-14-0-0-90:~$ kill -9 1546
ubuntu@ip-14-0-0-90:~$
```

Figura 3.3. Asesinato del proceso `1546` con `kill -9`

Se realiza la conexión con `sftp` para poder recuperar el archivo con `get`.

```
> sftp -i L01.pem ubuntu@100.29.110.48
Connected to 100.29.110.48.
sftp> ls
creausuarios.sh                                installnmon.log
ip-14-0-0-90_240924_0139.nmon                 schednmon.sh
sftp> get ip-14-0-0-90_240924_0139.nmon
Fetching /home/ubuntu/ip-14-0-0-90_240924_0139.nmon to ip-14-0-0-90_240924_0139.nmon
ip-14-0-0-90_240924_0139.nmon                  100%    75KB 197.4KB/s   00:00
sftp> ^D
> ls -l | grep "ip-14"
-rw-r--r-- 1 sn sn    76954 Sep 23 21:47 ip-14-0-0-90_240924_0139.nmon
> head -n 5 ip-14-0-0-90_240924_0139.nmon
AAA,progname,nmon
AAA,command,nmon -f -s 10 -c 60
AAA,version,16n
AAA,disks_per_line,150
AAA,max_disks,256,set by -d option

~/Downloads
```

Figura 3.4. Recuperación del archivo desde mi dispositivo local

Finalmente, se inspecciona el archivo con **NMON Visualizer** y se nota cómo hubo una gran carga de trabajo (debido al proceso con `cat`) hasta el momento en que se mató al proceso. Esto demuestra que `nmon` es una herramienta útil para monitorear el uso del CPU por los procesos en un sistema Linux.

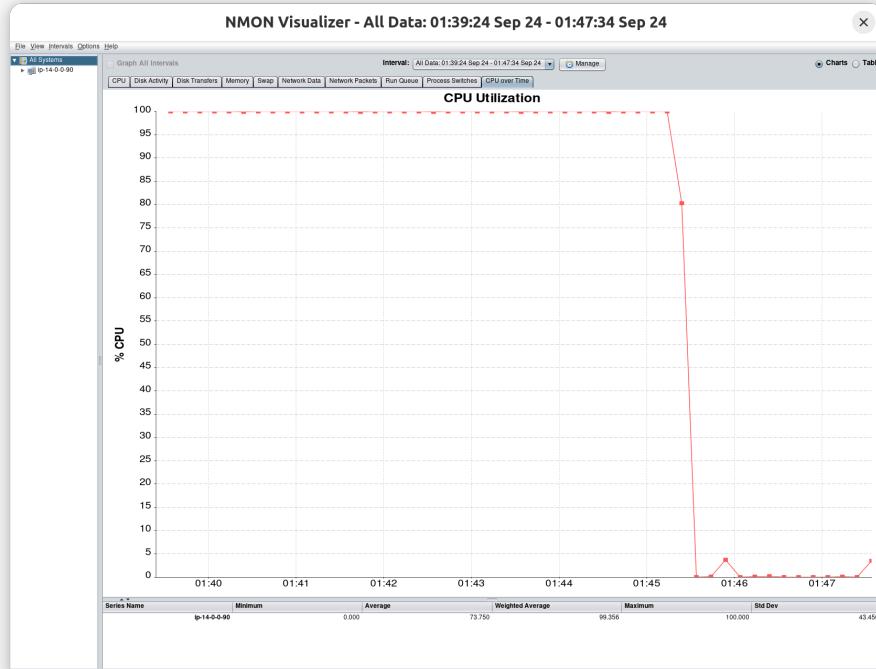
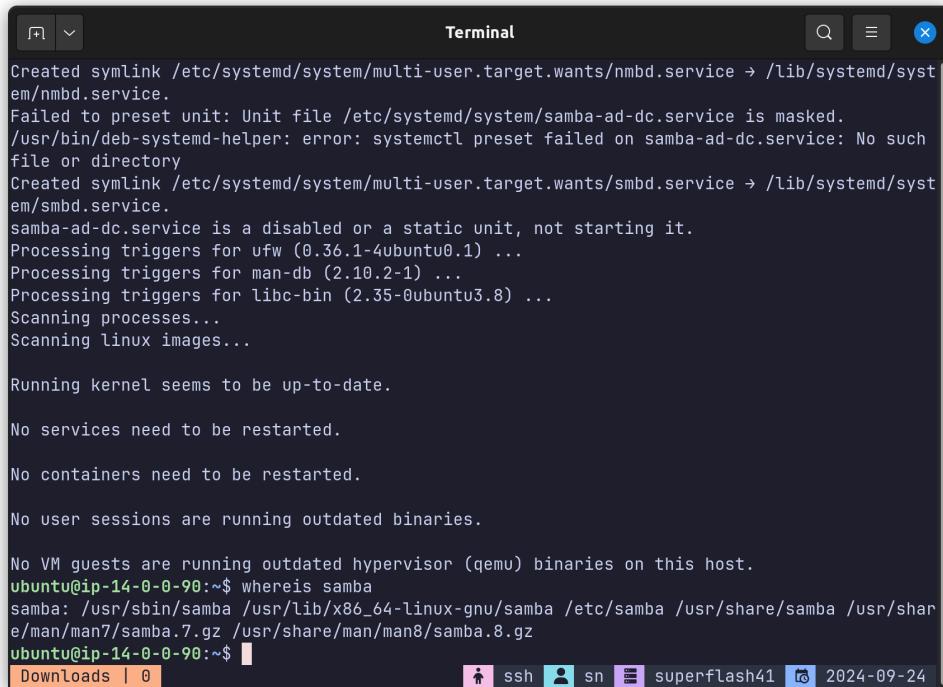


Figura 3.5. Inspección del archivo con **NMON Visualizer**

PREGUNTA 4

Se instala **samba** con el manejador de paquetes de Ubuntu.



```
Created symlink /etc/systemd/system/multi-user.target.wants/nmbd.service → /lib/systemd/system/nmbd.service.
Failed to preset unit: Unit file /etc/systemd/system/samba-ad-dc.service is masked.
/usr/bin/deb-systemd-helper: error: systemctl preset failed on samba-ad-dc.service: No such file or directory
Created symlink /etc/systemd/system/multi-user.target.wants/smbd.service → /lib/systemd/system/smbd.service.
samba-ad-dc.service is a disabled or a static unit, not starting it.
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

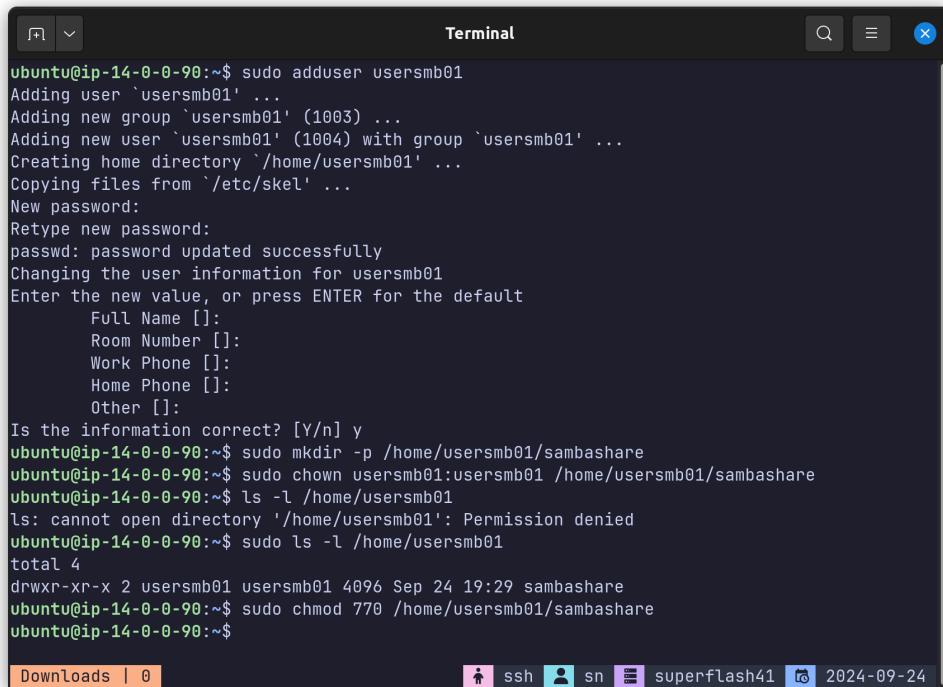
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-14-0-0-90:~$ whereis samba
samba: /usr/sbin/samba /usr/lib/x86_64-linux-gnu/samba /etc/samba /usr/share/samba /usr/share/man/man7/samba.7.gz /usr/share/man/man8/samba.8.gz
ubuntu@ip-14-0-0-90:~$
```

Figura 4.1. Instalación de **samba**

Se agrega al usuario **usersmb01** que tendrá el directorio a compartir por red.



```
ubuntu@ip-14-0-0-90:~$ sudo adduser usersmb01
Adding user `usersmb01' ...
Adding new group `usersmb01' (1003) ...
Adding new user `usersmb01' (1004) with group `usersmb01' ...
Creating home directory `/home/usersmb01' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for usersmb01
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []
Is the information correct? [Y/n] y
ubuntu@ip-14-0-0-90:~$ sudo mkdir -p /home/usersmb01/sambashare
ubuntu@ip-14-0-0-90:~$ sudo chown usersmb01:usersmb01 /home/usersmb01/sambashare
ubuntu@ip-14-0-0-90:~$ ls -l /home/usersmb01
ls: cannot open directory '/home/usersmb01': Permission denied
ubuntu@ip-14-0-0-90:~$ sudo ls -l /home/usersmb01
total 4
drwxr-xr-x 2 usersmb01 usersmb01 4096 Sep 24 19:29 sambashare
ubuntu@ip-14-0-0-90:~$ sudo chmod 770 /home/usersmb01/sambashare
ubuntu@ip-14-0-0-90:~$
```

Figura 4.2. Creación del directorio **sambashare**

```

path = /var/spool/samba
printable = yes
guest ok = no
read only = yes
create mask = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin

[sambashare]
comment = L03 Sambaaaa
path = /home/usersmb01/sambashare
read only = no
browsable = yes

"/etc/samba/smb.conf" 247L, 9055B written

```

247,0-1 Bot
Downloads | 0 ssh sn superflash41 2024-09-24

Figura 4.3. Configuración de parámetros para el directorio **sambashare**

```

# sending the correct chat script for the passwd program in Debian Sarge).
passwd program = /usr/bin/passwd %
passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n *password\
supdated\ssuccessfully*.

# This boolean controls whether PAM will be used for password changes
# when requested by an SMB client instead of the program listed in
# 'passwd program'. The default is 'no'.
pam password change = yes

# This option controls how unsuccessful authentication attempts are mapped
# to anonymous connections
#   map to guest = bad user

##### Domains #####
#
# The following settings only takes effect if 'server role = classic'
# primary domain controller', 'server role = classic backup domain controller'
# or 'domain logons' is set
#
# It specifies the location of the user's
# profile directory from the client point of view) The following
# required a [profiles] share to be setup on the samba server (see
# below)
"/etc/samba/smb.conf" 247L, 9056B written

```

100,4 39%
Downloads | 0 ssh sn superflash41 2024-09-24

Figura 4.4. Se descomenta el parámetro **map to guest**

```
ubuntu@ip-14-0-0-90:~$ sudo vim /etc/samba/smb.conf
ubuntu@ip-14-0-0-90:~$ sudo service smbd restart
ubuntu@ip-14-0-0-90:~$ sudo ufw allow samba
Rules updated
Rules updated (v6)
ubuntu@ip-14-0-0-90:~$ sudo smbpasswd -a usersmb01
New SMB password:
Retype new SMB password:
Added user usersmb01.
ubuntu@ip-14-0-0-90:~$
```

Downloads | 0 ssh sn superflash41 2024-09-24

Figura 4.5. Reinicio del servicio y configuración de contraseña

Luego, se crea una regla de entrada desde la consola de AWS para que se pueda realizar la conexión de tipo **SMB**.

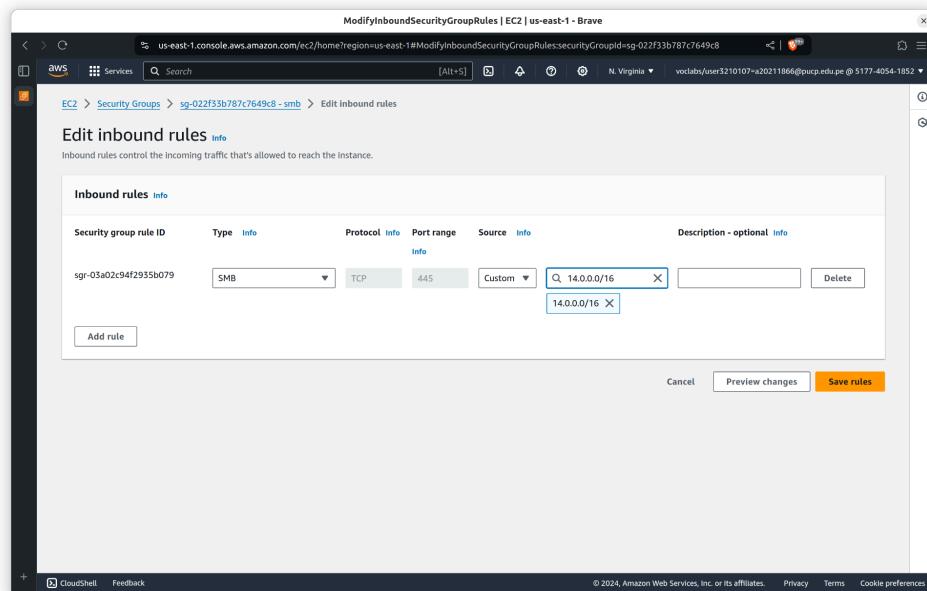


Figura 4.6. Creación de una regla de entrada para instancia Linux

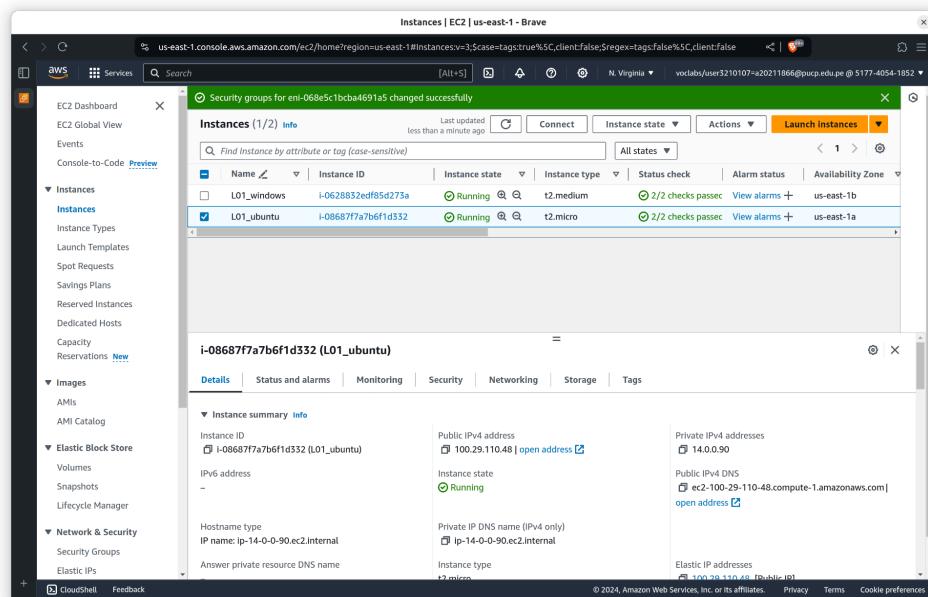


Figura 4.7. Regla de entrada añadida

Finalmente, desde la instancia Windows se accede a la dirección `\\"14.0.0.90\sambashare` y se ingresan las credenciales antes configuradas.

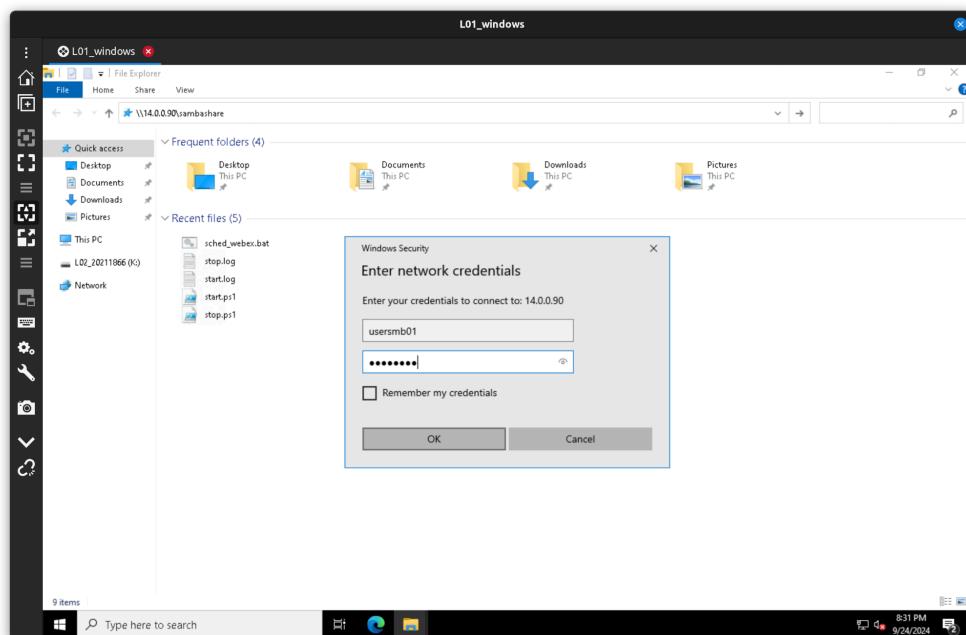


Figura 4.8. Ingreso de credenciales de acceso

Como se muestra a continuación, se crea un directorio llamado `L03` con un archivo en el directorio compartido.

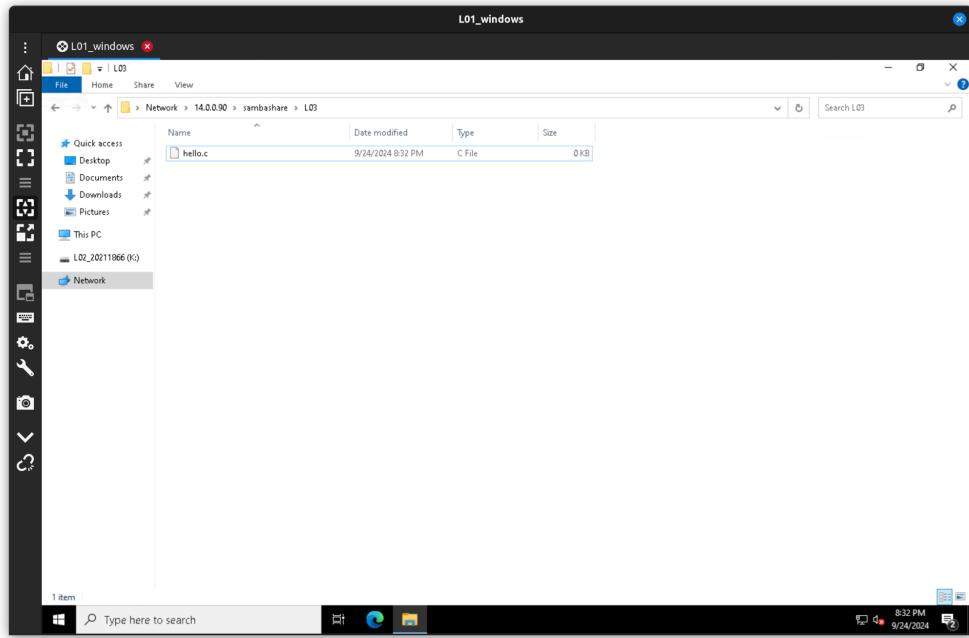


Figura 4.9. Creacion de archivo en el directorio L03

Se puede verificar la creación del directorio desde la instancia Linux.

```
ubuntu@ip-14-0-0-90:~$ sudo ls -l /home/usersmb01/sambashare/
total 4
drwxr-xr-x 2 usersmb01 usersmb01 4096 Sep 24 20:32 L03
ubuntu@ip-14-0-0-90:~$ sudo ls -l /home/usersmb01/sambashare/L03/
total 0
-rwxr--r-- 1 usersmb01 usersmb01 0 Sep 24 20:32 hello.c
ubuntu@ip-14-0-0-90:~$
```

Figura 4.10. Directorio L03 visto desde la instancia Linux

PREGUNTA 5

Se accede a la configuración de Windows para la creación de los grupos y usuarios.

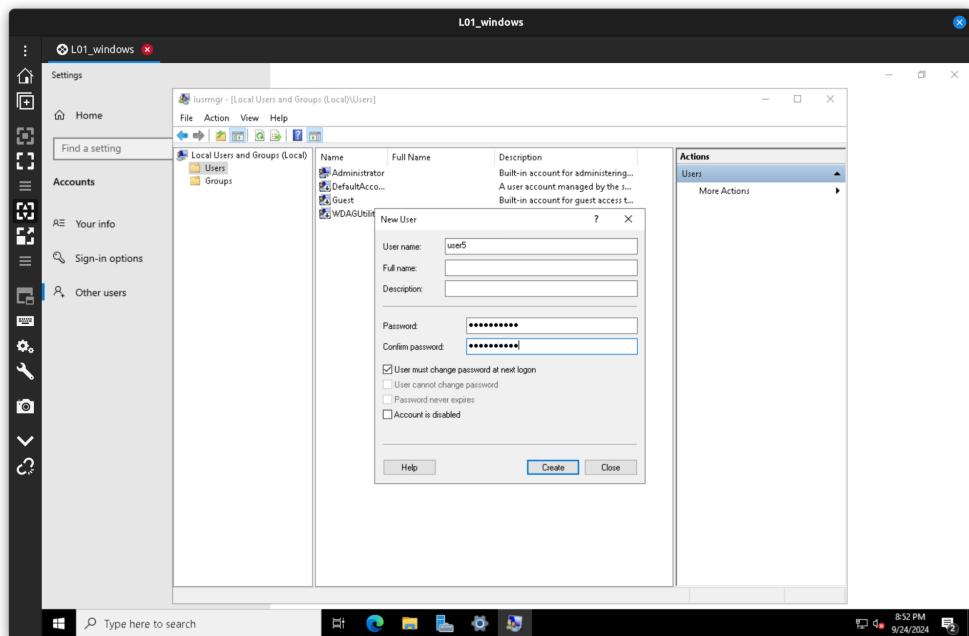


Figura 5.1. Creación de los usuarios

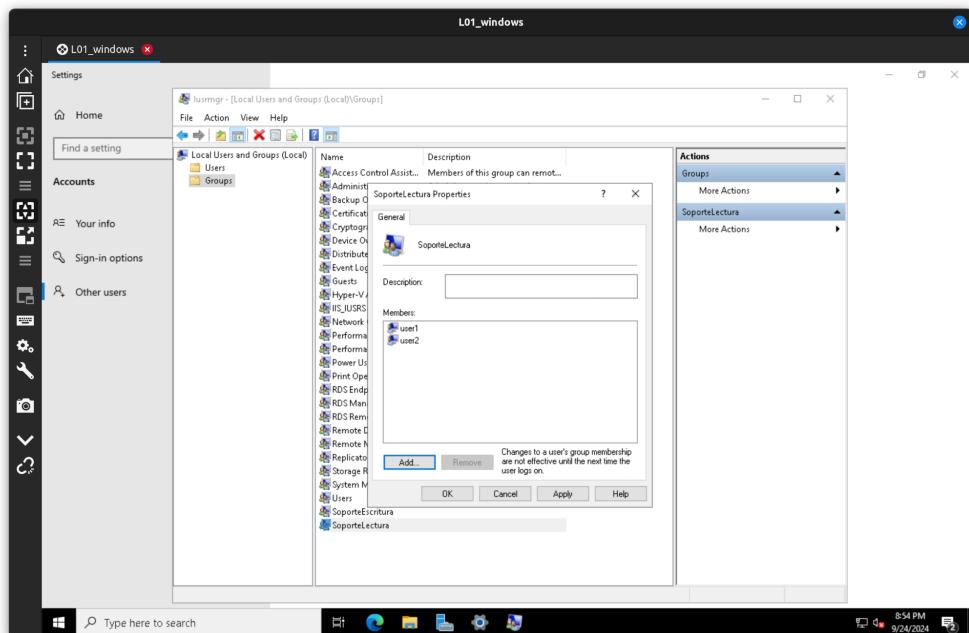


Figura 5.2. Creación del grupo SoporteLectura

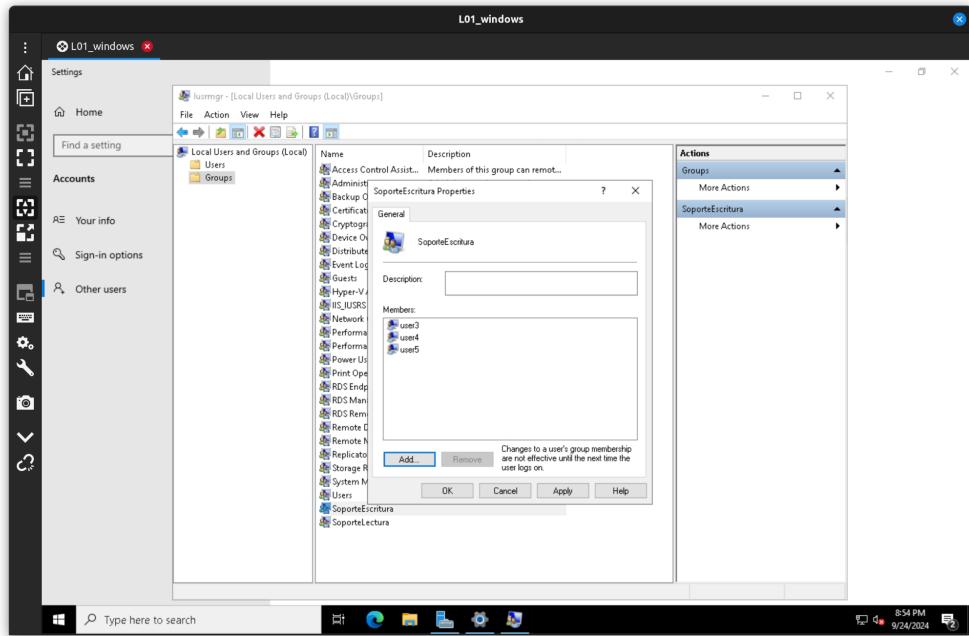


Figura 5.3. Creación del grupo SoporteEscritura

Se crea la carpeta Apps en el disco local C::.

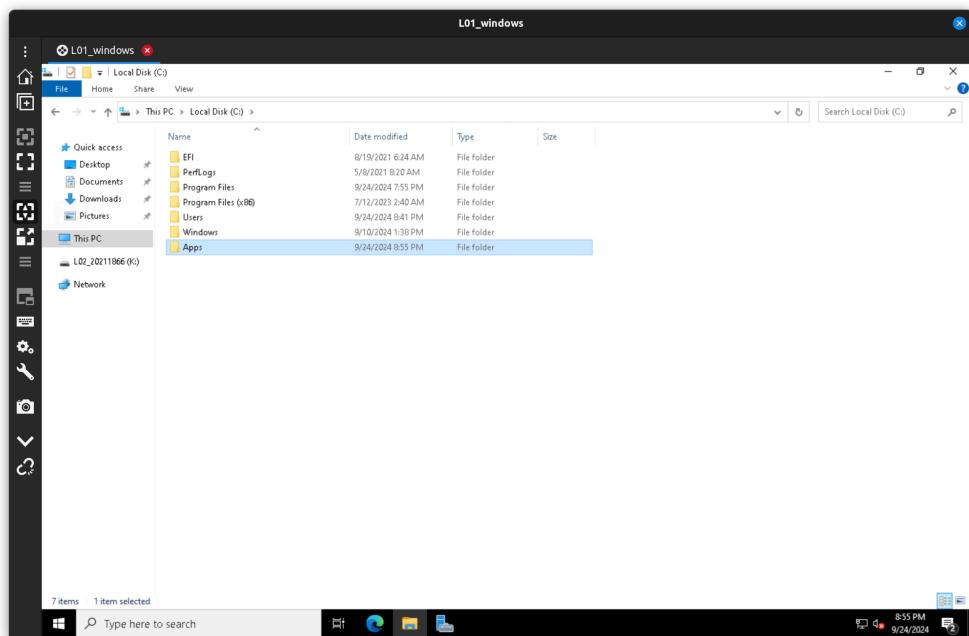


Figura 5.4. Creación de la carpeta a compartir

Luego, se otorgan permisos de escritura y lectura a los grupos de acuerdo a su rol.

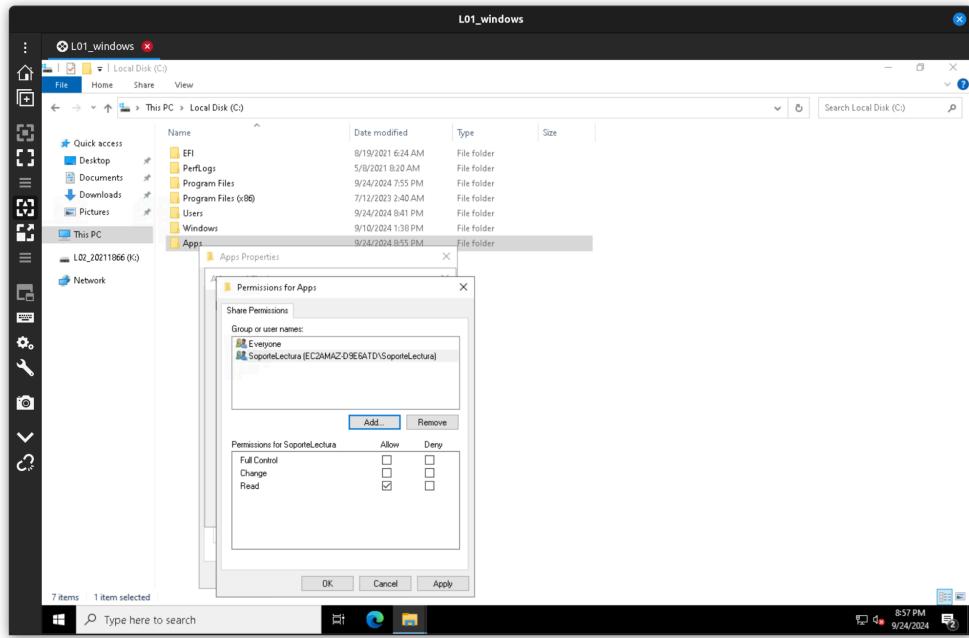


Figura 5.5. Asignación de permisos al grupo SoporteLectura

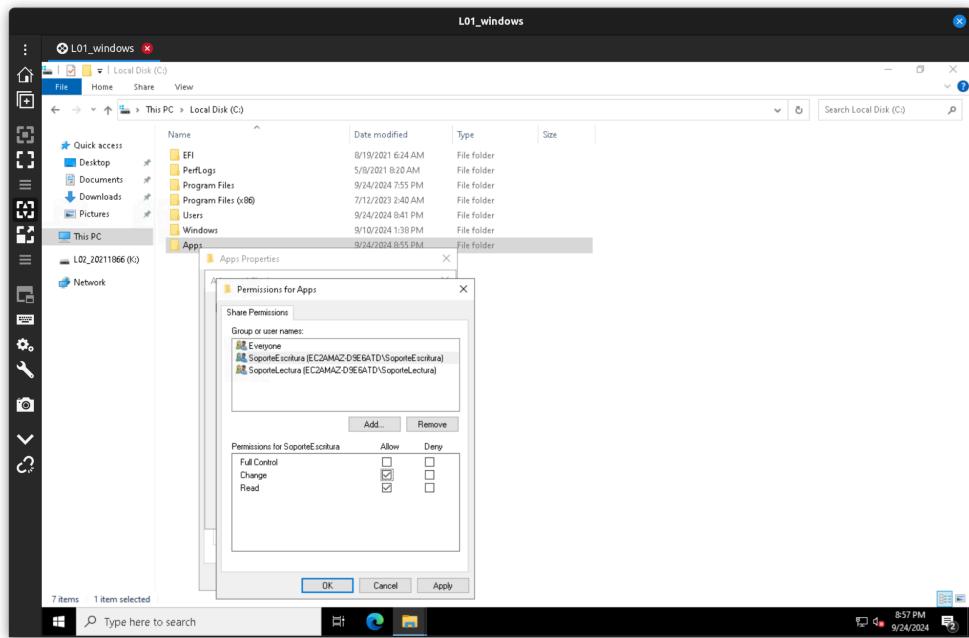


Figura 5.6. Asignación de permisos al grupo SoporteEscritura

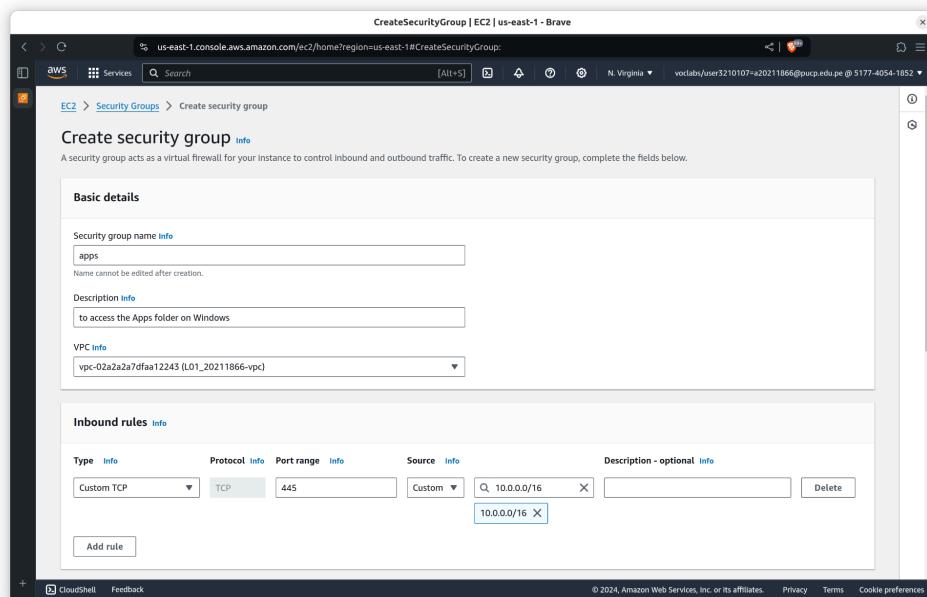


Figura 5.7. Creación de una regla de entrada para instancia Windows

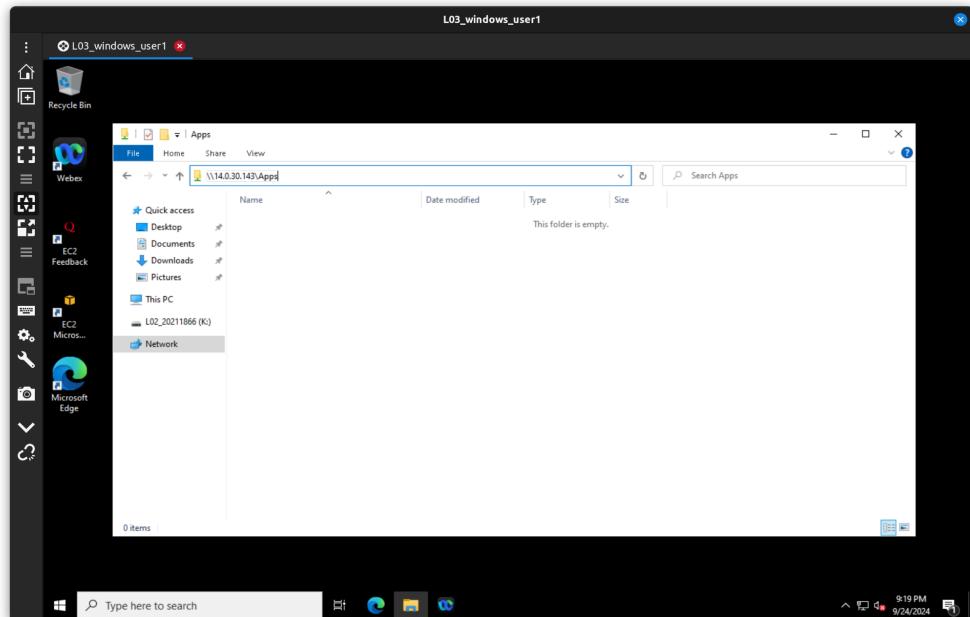


Figura 5.8. Acceso a la carpeta compartida por el usuario **user01**

Por un lado, como el usuario **user01** pertenece al grupo de lectura, no puede crear un archivo en el directorio compartido. Esto queda demostrado en la siguiente imagen donde se muestra un mensaje de error al intentar crear un archivo de texto.

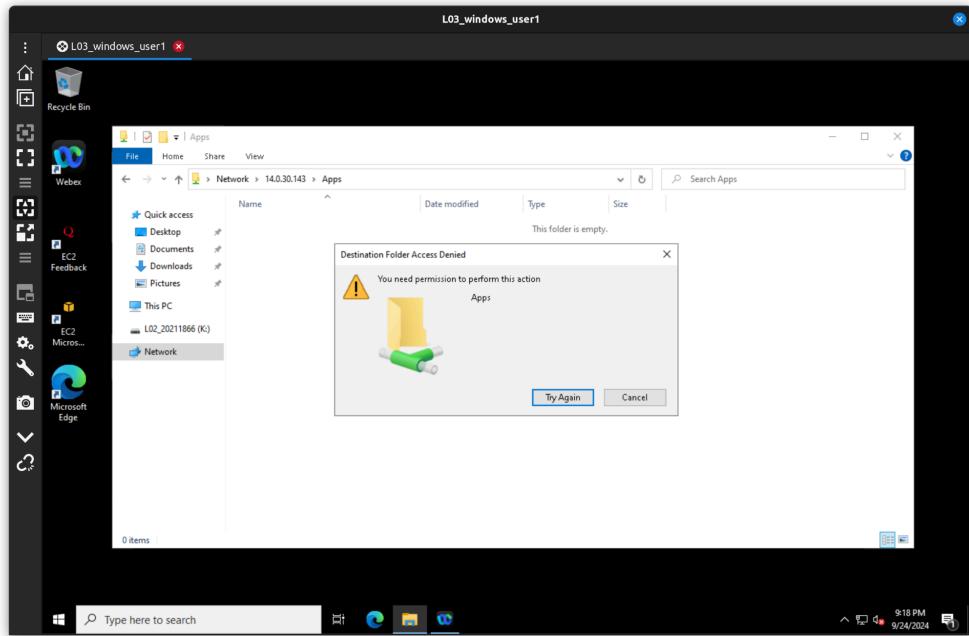


Figura 5.9. Impedimento de creación de un archivo para **user01**

Por otro lado, dado que el usuario **user05** pertenece al grupo de escritura, sí tiene permiso para crear el archivo en el directorio compartido.

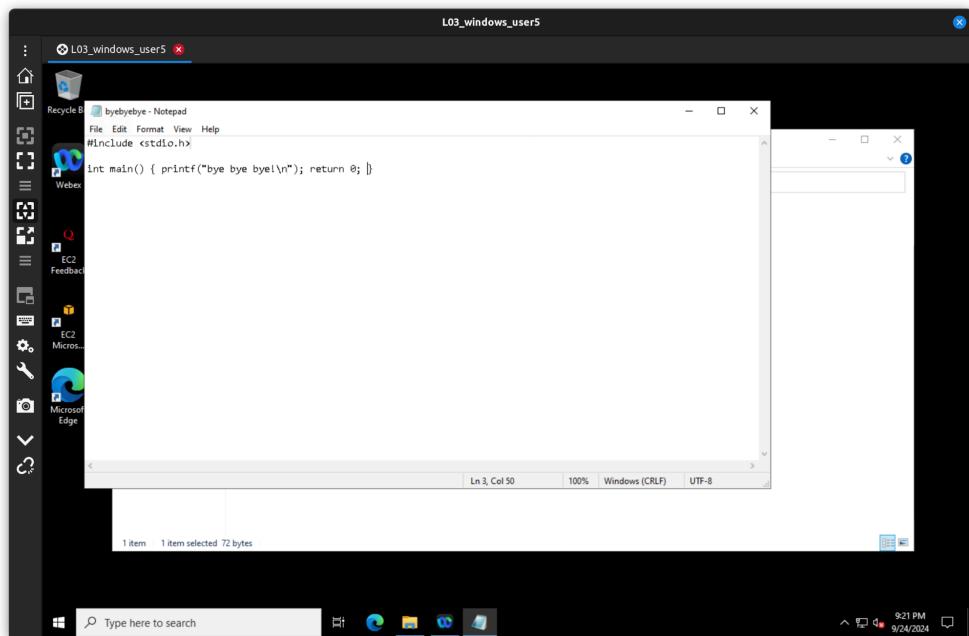


Figura 5.10. Permiso de creación de un archivo para **user05**

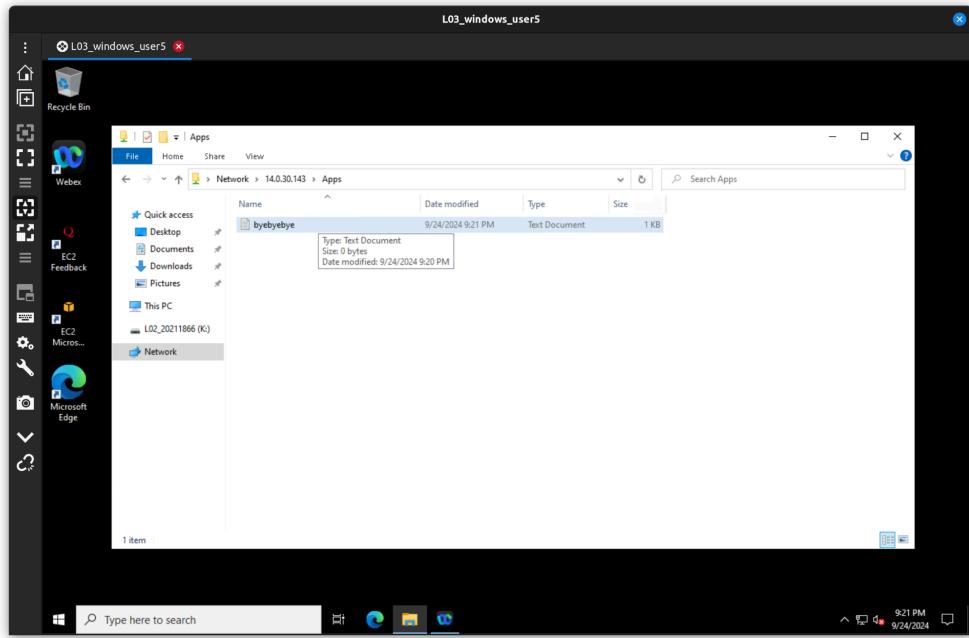
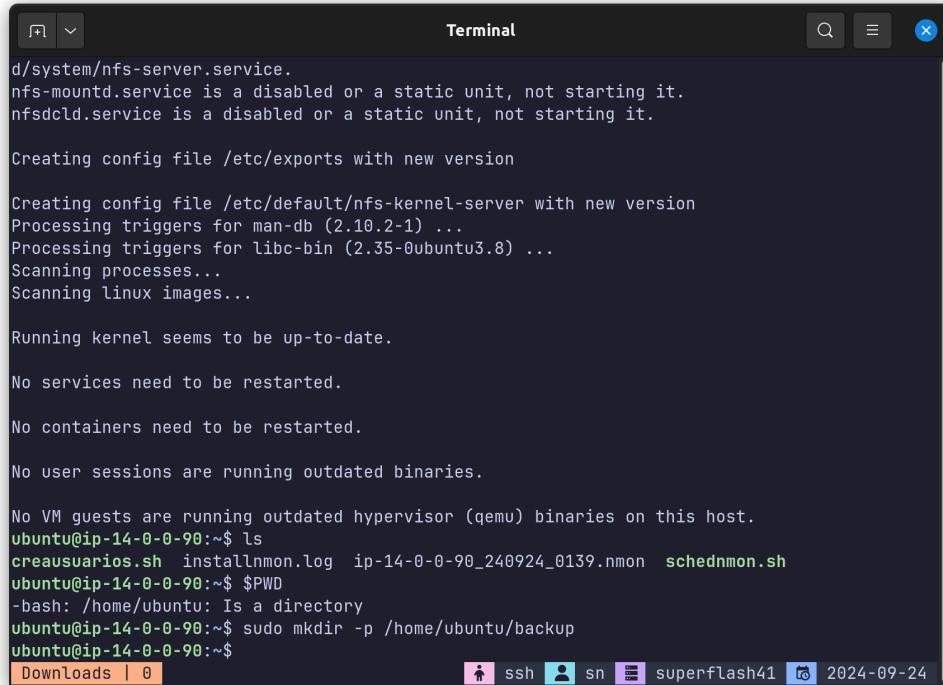


Figura 5.11. Archivo creado por user05

PREGUNTA 6

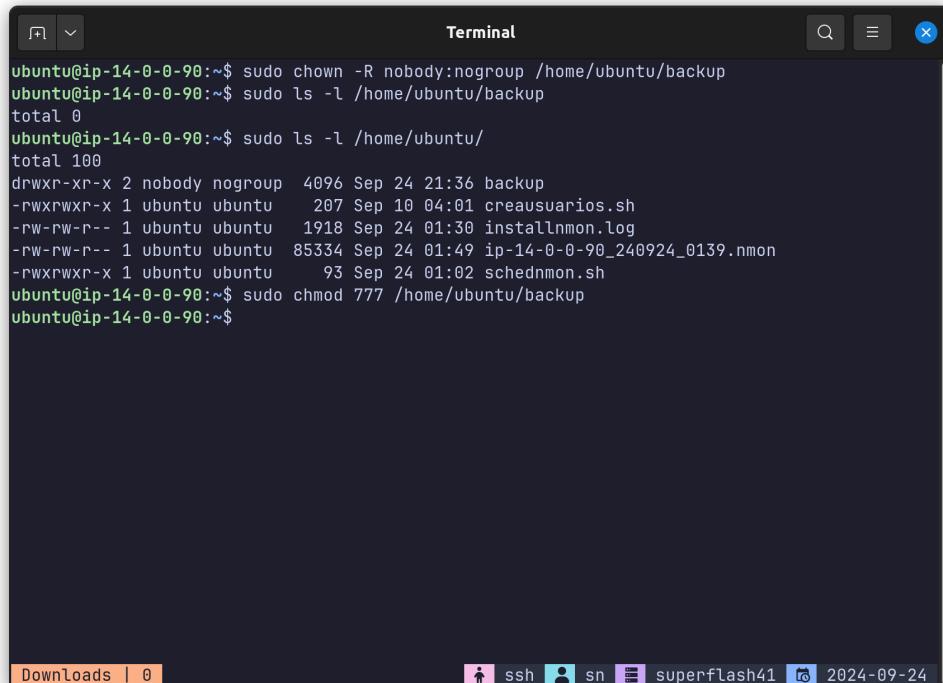
Se instala la utilidad **nfs-kernel-server** con el manejador de maquetes de Ubuntu.



```
d/system/nfs-server.service.  
nfs-mountd.service is a disabled or a static unit, not starting it.  
nfsdcl.d.service is a disabled or a static unit, not starting it.  
  
Creating config file /etc/exports with new version  
  
Creating config file /etc/default/nfs-kernel-server with new version  
Processing triggers for man-db (2.10.2-1) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-14-0-0-90:~$ ls  
creausuarios.sh installnmon.log ip-14-0-0-90_240924_0139.nmon schednmon.sh  
ubuntu@ip-14-0-0-90:~$ $PWD  
-bash: /home/ubuntu: Is a directory  
ubuntu@ip-14-0-0-90:~$ sudo mkdir -p /home/ubuntu/backup  
ubuntu@ip-14-0-0-90:~$  
Downloads | 0    ssh sn superflash41 2024-09-24
```

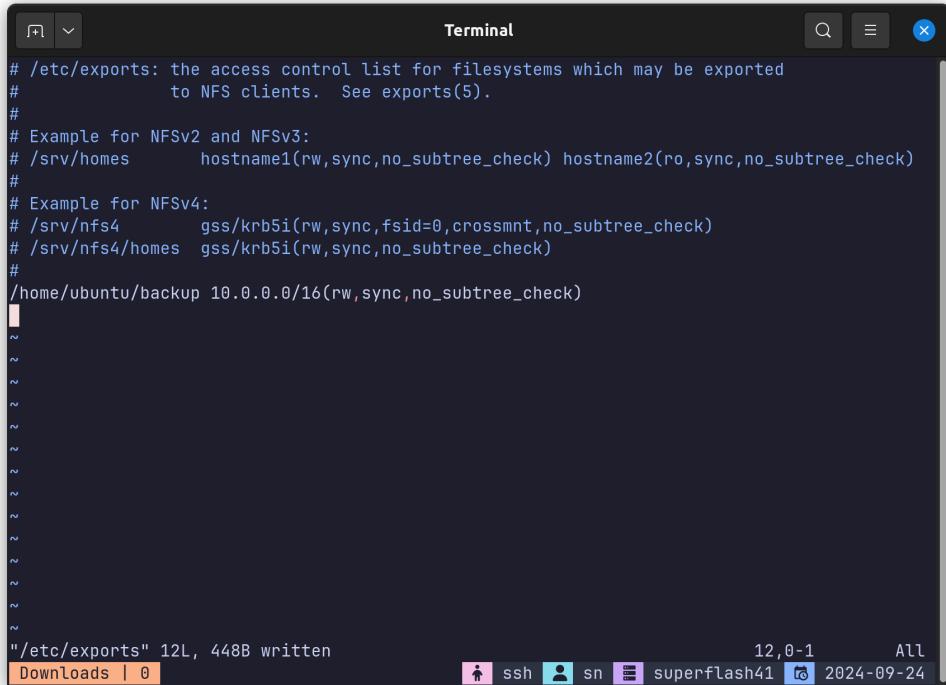
Figura 6.1. Creación del directorio **backup** a compartir

Se cambian los permisos de acceso al directorio.



```
ubuntu@ip-14-0-0-90:~$ sudo chown -R nobody:nogroup /home/ubuntu/backup  
ubuntu@ip-14-0-0-90:~$ sudo ls -l /home/ubuntu/backup  
total 0  
ubuntu@ip-14-0-0-90:~$ sudo ls -l /home/ubuntu/  
total 100  
drwxr-xr-x 2 nobody nogroup 4096 Sep 24 21:36 backup  
-rwxrwxr-x 1 ubuntu ubuntu 207 Sep 10 04:01 creausuarios.sh  
-rw-rw-r-- 1 ubuntu ubuntu 1918 Sep 24 01:30 installnmon.log  
-rw-rw-r-- 1 ubuntu ubuntu 85334 Sep 24 01:49 ip-14-0-0-90_240924_0139.nmon  
-rwxrwxr-x 1 ubuntu ubuntu 93 Sep 24 01:02 schednmon.sh  
ubuntu@ip-14-0-0-90:~$ sudo chmod 777 /home/ubuntu/backup  
ubuntu@ip-14-0-0-90:~$  
Downloads | 0    ssh sn superflash41 2024-09-24
```

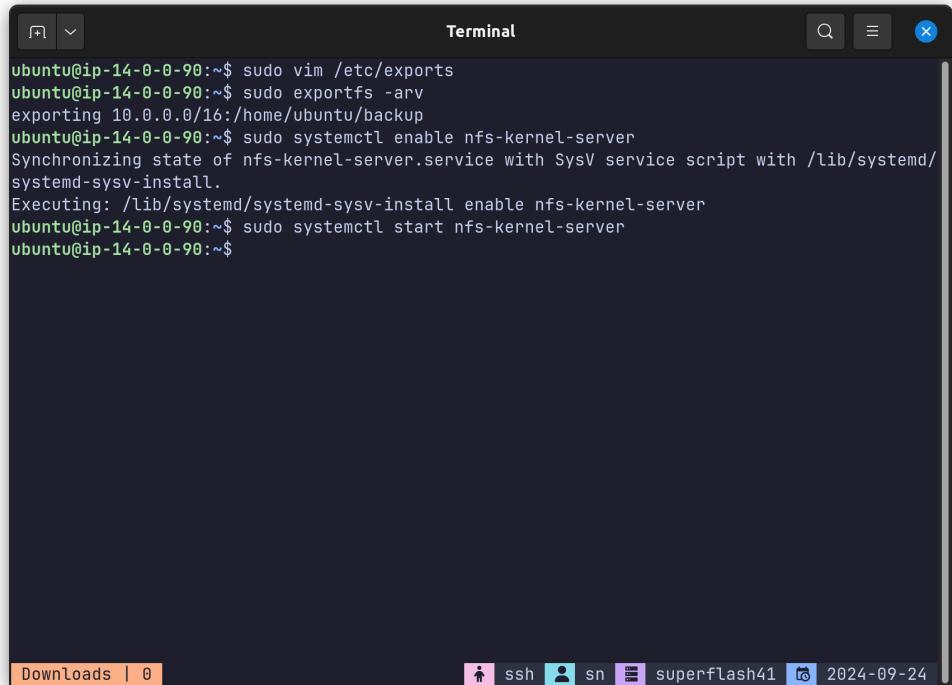
Figura 6.2. Cambio de permisos del directorio **/home/ubuntu/backup**



```
# /etc/exports: the access control list for filesystems which may be exported
#           to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4      gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home/ubuntu/backup 10.0.0.0/16(rw,sync,no_subtree_check)

"/etc/exports" 12L, 448B written
```

Figura 6.3. Se comparte el directorio con otros miembros de la subnet privada

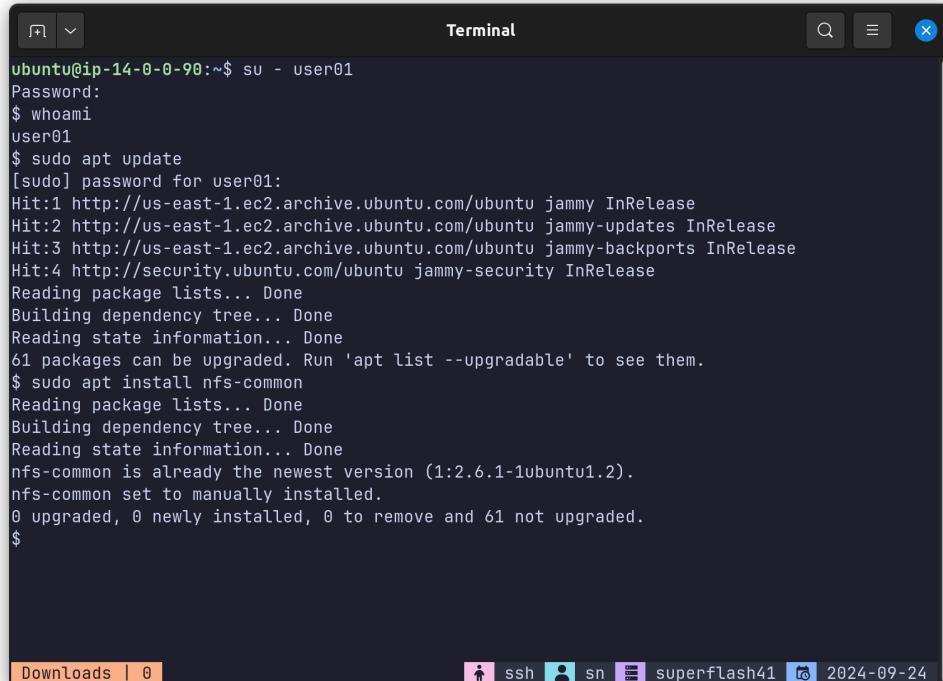


```
ubuntu@ip-14-0-0-90:~$ sudo vim /etc/exports
ubuntu@ip-14-0-0-90:~$ sudo exportfs -arv
exporting 10.0.0.0/16:/home/ubuntu/backup
ubuntu@ip-14-0-0-90:~$ sudo systemctl enable nfs-kernel-server
Synchronizing state of nfs-kernel-server.service with SysV service script with /lib/systemd/
systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nfs-kernel-server
ubuntu@ip-14-0-0-90:~$ sudo systemctl start nfs-kernel-server
ubuntu@ip-14-0-0-90:~$
```

Figura 6.4. Reinicio del servicio

Es importante mencionar que el objetivo de la actividad es simular cómo un directorio se puede compartir entre un servidor y clientes con el sistema de archivos NFS. En este caso, se usa al usuario **ubuntu** como servidor y a los usuarios **user01** y **user02** como clientes. De esta forma, cada uno tiene la copia de respaldo y el servidor puede acceder a

ella. El montaje del directorio se realizará en cada cliente a nivel de sus propios subdirectorios simulando que cada uno pertenece a un nodo distinto de una red.

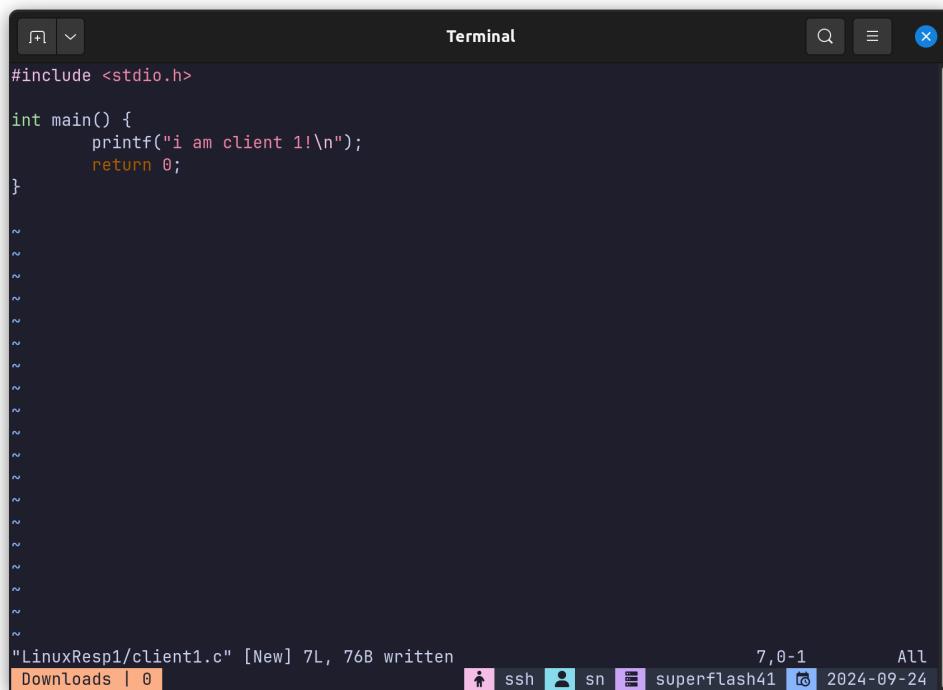


The screenshot shows a terminal window titled "Terminal". The command entered is "sudo apt update", which lists several package sources and their versions. Then, "sudo apt install nfs-common" is run, showing that the package is already at its newest version (1:2.6.1-1ubuntu1.2) and has been manually installed. The terminal window has a dark background with light-colored text. The bottom status bar shows icons for user, ssh, sn, and superflash41, along with the date 2024-09-24.

```
ubuntu@ip-14-0-0-90:~$ su - user01
Password:
$ whoami
user01
$ sudo apt update
[sudo] password for user01:
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
61 packages can be upgraded. Run 'apt list --upgradable' to see them.
$ sudo apt install nfs-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-common is already the newest version (1:2.6.1-1ubuntu1.2).
nfs-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 61 not upgraded.
$
```

Figura 6.5. Cliente 1 instala **nfs-common**

Como en nuestro caso el servidor y los clientes pertenecen a un mismo sistema, la utilidad ya se encontraba instalada (en la realidad cada cliente debe instalar la utilidad). Luego, se procede a montar el directorio con la ip compartida y se crea el subdirectorio para la carpeta de respaldo.



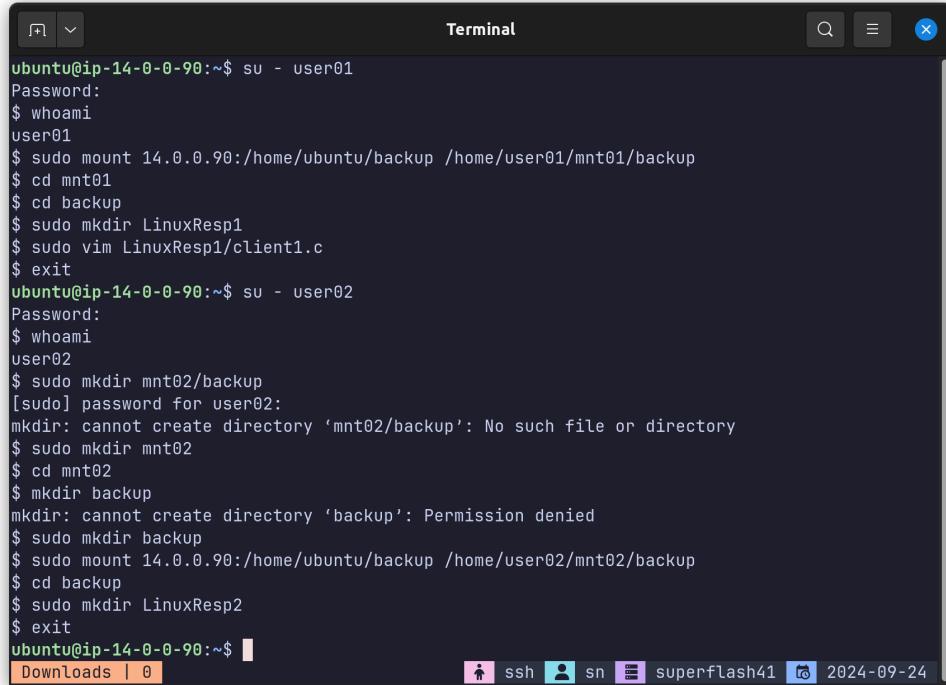
The screenshot shows a terminal window titled "Terminal". The command entered is "cat > client1.c", followed by the code for a simple C program that prints "I am client 1!". The file is saved with the command "ctrl+D". The terminal window has a dark background with light-colored text. The bottom status bar shows icons for user, ssh, sn, and superflash41, along with the date 2024-09-24.

```
#include <stdio.h>

int main() {
    printf("I am client 1!\n");
    return 0;
}

"LinuxResp1/client1.c" [New] 7L, 76B written
Downloads | 0
```

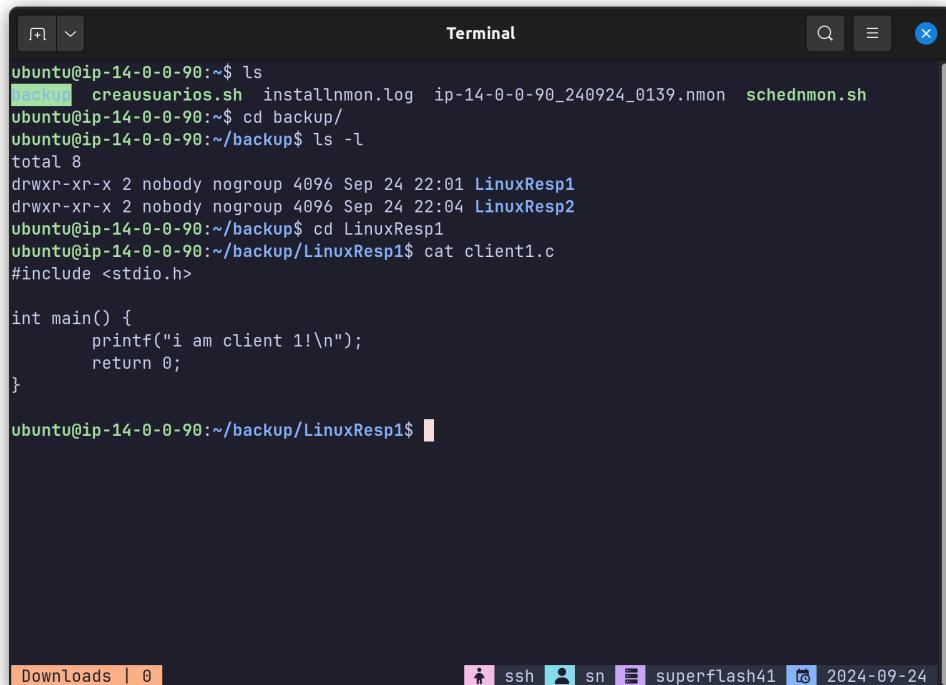
Figura 6.6. Archivo creado por el cliente 1 en LinuxResp1



```
ubuntu@ip-14-0-0-90:~$ su - user01
Password:
$ whoami
user01
$ sudo mount 14.0.0.90:/home/ubuntu/backup /home/user01/mnt01/backup
$ cd mnt01
$ cd backup
$ sudo mkdir LinuxResp1
$ sudo vim LinuxResp1/client1.c
$ exit
ubuntu@ip-14-0-0-90:~$ su - user02
Password:
$ whoami
user02
$ sudo mkdir mnt02/backup
[sudo] password for user02:
mkdir: cannot create directory 'mnt02/backup': No such file or directory
$ sudo mkdir mnt02
$ cd mnt02
$ mkdir backup
mkdir: cannot create directory 'backup': Permission denied
$ sudo mkdir backup
$ sudo mount 14.0.0.90:/home/ubuntu/backup /home/user02/mnt02/backup
$ cd backup
$ sudo mkdir LinuxResp2
$ exit
ubuntu@ip-14-0-0-90:~$
```

Figura 6.7. Montaje y creación de archivos por cada cliente

Finalmente, se puede acceder a ambias carpetas de respaldo desde el servidor.



```
ubuntu@ip-14-0-0-90:~$ ls
[backu] creausuarios.sh installnmon.log ip-14-0-0-90_240924_0139.nmon schednmon.sh
ubuntu@ip-14-0-0-90:~$ cd backup/
ubuntu@ip-14-0-0-90:~/backup$ ls -l
total 8
drwxr-xr-x 2 nobody nogroup 4096 Sep 24 22:01 LinuxResp1
drwxr-xr-x 2 nobody nogroup 4096 Sep 24 22:04 LinuxResp2
ubuntu@ip-14-0-0-90:~/backup$ cd LinuxResp1
ubuntu@ip-14-0-0-90:~/backup/LinuxResp1$ cat client1.c
#include <stdio.h>

int main() {
    printf("i am client 1!\n");
    return 0;
}

ubuntu@ip-14-0-0-90:~/backup/LinuxResp1$
```

Figura 6.8. Acceso a carpetas de respaldo desde el servidor