

5.1.Utils_Manager_Objects_Domain

Emilio Cabo Gomis

Índice

LDB.....	2
Python.....	6
Comprobaciones.....	9

5.1 UTILS MANAGER

Durante las navidades de el año 2024 – 2025 refiné mis conocimientos acerca de la linea e intérpete de comandos del Linux BASH mediante la lectura del libro [The Linux Command Line](#). Además mediante recursos online he empezado a aprender a programar en python. Dado que es un lenguaje basado en C es algo similar a este, por lo que me han resultado muy útiles los conocimientos previos.

LDB

Para la automatización de la carga de usuarios en el examen he escrito una herramienta en bash que lee de un archivo CSV los usuarios y los grupos del examen.

El archivo CSV fuente es el siguiente:

```
usuario, uid, grupo
noa04,10000,"g_VENTAS0"
fer04,11000,"g_VENTAS0"
noa04,10000,"g_VENTAL1"
ana04,18000,"g_VENTAL1"
felix04,12000,"g_VENTAE1"
beatriz04,19000,"g_VENTAE1"
manuel04,13000,"g_ASIA2"
geluco04,16000,"g_ASIA2"
fer04,11000,"g_EUROPA2"
honorato04,17000,"g_EUROPA2"
```

Los que tengáis un ojo avisado os daréis cuenta que el archivo tiene el tercer campo del CSV encapsulado en comillas dobles (""); esto tiene una función: En caso de este documento ser editado por un editor de textos programado para Windows. Este puede introducir caracteres de retorno de carro no deseados, tabuladores u otros caracteres no visibles que rompan la funcionalidad del script. Para evitar estos problemas hay una sección del script que reescribe la variable del tercer campo para incluir solo lo encapsulado entre las comillas.

Esta sección es la siguiente: `nomGrupo=`echo "$nomGrupo" | awk -F'"' '{print $2}``

Las variables de este script son las siguientes.

5.1 UTILS MANAGER

```
#variables
INPUT=estructura/j01_groupusr_demanda.csv
OUTPUT_G=ldif/j01_grupos_demanda.ldif
OUTPUT_U=ldif/j01_usuarios_demanda.ldif
#OJO: la variable del los grupos virgenes es 10000 por eso empleamos 11000
#estos pueden estar entre el 10000 - 999999
gid=11000

directory_check ldif
removeLdif
saltar_encabezado=1
echo 'añadiendo los grupos y ous demandados en el examen'
createLdif $OUTPUT_G
```

La sección de código que se dedica a ejecutar las funciones de este script es la siguiente.

```
OLDIFS=$IFS
IFS=', '
[ ! -f $INPUT ] && { echo "$INPUT file not found"; exit 99; }
while read nomUsuario uid nomGrupo
do
    if ((saltar_encabezado)); then
        ((saltar_encabezado--))
    else
        nomGrupo=`echo "$nomGrupo" | awk -F'|' '{print $2}'`
        echo "Línea procesada: $nomUsuario, $uid, $nomGrupo"
        dn_grupo="cn=${nomGrupo},ou=grupos,dc=$KRBORG,dc=$KRBEND"
        dn_usuario="cn=${nomUsuario},ou=usuarios,dc=$KRBORG,dc=$KRBEND"
        groupCheck $nomGrupo $OUTPUT_G
        if [[ $? -ne 0 ]] ; then
            createGroup "$dn_grupo" "$gid" "$nomGrupo"
            gid=$((gid+1))
        else
            echo "grupo $nomGrupo ya existe"
        fi

        #comprobamos si el usuario existe
        userCheck $nomUsuario $OUTPUT_U
        if [[ $? -ne 0 ]] ; then
            createUser "$dn_usuario" "$nomUsuario" "$uid" "$gid"
        fi
        echo "DATOS GRUPO"
        echo "nombregrupo : $nomGrupo"
        echo "dngrupo : $dn_grupo"
        echo "gid : $gid"
        echo "DATOS USUARIO"
        echo "nombreusuario : $nomUsuario"
        echo "dn_usuario : $dn_usuario"
    fi
done < $INPUT
IFS=$OLDIFS

apt-get install ldb-tools -y

#Importacion de los grupos
ldbadd --url=/var/lib/samba/private/sam.ldb $OUTPUT_G
```

El comportamiento de este script es leer del archivo CSV cuales son los usuarios que vamos a crear, comprobar que no existen ya en el sistema y crearlo. Este script repite el proceso también para los grupos.

5.1 UTILS MANAGER

Si probamos a ejecutarlo:

Primero nos imprime en el terminal la linea que ha procesado y los códigos de retorno de algunas funciones.

```
Windows PowerShell
dnsgupo : cn=g_ASIA2,ou=grupos,dc=lpyme10,dc=net
gid : 11800
DATOS USUARIO
nombreusuario : manuel04
dn_usuario : cn=manuel04,ou=usuarios,dc=lpyme10,dc=net
Línea procesada: geluco04, 16000, g_ASIA2
g_ASIA2
nomGrupo g_ASIA2 encontrado no es necesario añadirlo
grupo g_ASIA2 ya existe
usuario geluco04 no encontrado, llamando a funcion añadir
DATOS GRUPO
nombregrupo : g_ASIA2
dngrupo : cn=g_ASIA2,ou=grupos,dc=lpyme10,dc=net
gid : 11800
DATOS USUARIO
nombreusuario : geluco04
dn_usuario : cn=geluco04,ou=usuarios,dc=lpyme10,dc=net
Línea procesada: fer04, 11800, g_EUROPA2
g_EUROPA2
nomGrupo g_EUROPA2 encontrado no es necesario añadirlo
grupo g_EUROPA2 ya existe
DATOS GRUPO
nombregrupo : g_EUROPA2
dngrupo : cn=g_EUROPA2,ou=grupos,dc=lpyme10,dc=net
gid : 11800
DATOS USUARIO
nombreusuario : fer04
dn_usuario : cn=fer04,ou=usuarios,dc=lpyme10,dc=net
Línea procesada: honorato04, 17000, g_EUROPA2
g_EUROPA2
nomGrupo g_EUROPA2 encontrado no es necesario añadirlo
grupo g_EUROPA2 ya existe
usuario honorato04 no encontrado, llamando a funcion añadir
DATOS GRUPO
nombregrupo : g_EUROPA2
dngrupo : cn=g_EUROPA2,ou=grupos,dc=lpyme10,dc=net
gid : 11800
DATOS USUARIO
nombreusuario : honorato04
dn_usuario : cn=honorato04,ou=usuarios,dc=lpyme10,dc=net
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ldb-tools ya está en su versión más reciente (2:2.2.3-2-deb11u2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Added 0 records successfully
Added 8 records successfully
archivo ldif/*.txt encontrado, eliminando
rm: no se puede borrar 'ldif/*.txt': No existe el fichero o el directorio
Directorio ldif limpio y listo para trabajo
ldif/tmp_usuariosusados.txt no existe, creando...
```

Luego crea los LDIF

```
File Edit Selection View Go Run Terminal Help
HERREMIENTAS_NAVIDAD
explorer
conf_files
estructura
ldif
j01_grupos_demanda.ldif
j01_usuarios_demanda.ldif
tmp_usuariosusados.txt
pbis-open-9.1.0.551.linux.x86_64.deb
00_variables_locales.sh
01_red_debian.sh
02_redubuntu.sh
05_versionDebian.sh
06_crearusrLocales.sh
07_D_samba4.sh
08_D_promos4.sh
09_D_comprobaciones.sh
10_D_DHCP.sh
11_U_unirUbuntu.sh
12_4_D_ou_grupos_migrash
13_3_D_ou_demanda.sh
14_2_D_grupos_demanda.sh
16_D_format.sh
17_3_D_esqueleto.sh
18_D_nslcd_deb.sh
19_2_D_samba_share.sh
20_D_ACL.sh
50_mpre_lpr_cabgomei.sh
51_mpre_lpr_cabgomei.sh
100_D_copiasseguridadconforntab.sh
101_D_copiasseguridadconforntab.sh
102_D_copiasseguridadconforntab.sh
110_D_copiasseguridadconforntab.sh
122_4_D_cargaMasiva.sh
999_sshSetup_cabgomei.sh
addubuntu.sh
guia_modif_examenes.txt
guia_samba_navidad.odt
history_debandidirectoriotxt
history_directoriocomun.txt
history_directoriohome.txt
history_VUbuntu.txt
OUTLINE
TIMELINE
Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text
10:28
13/01/2025
```

5.1 UTILS MANAGER

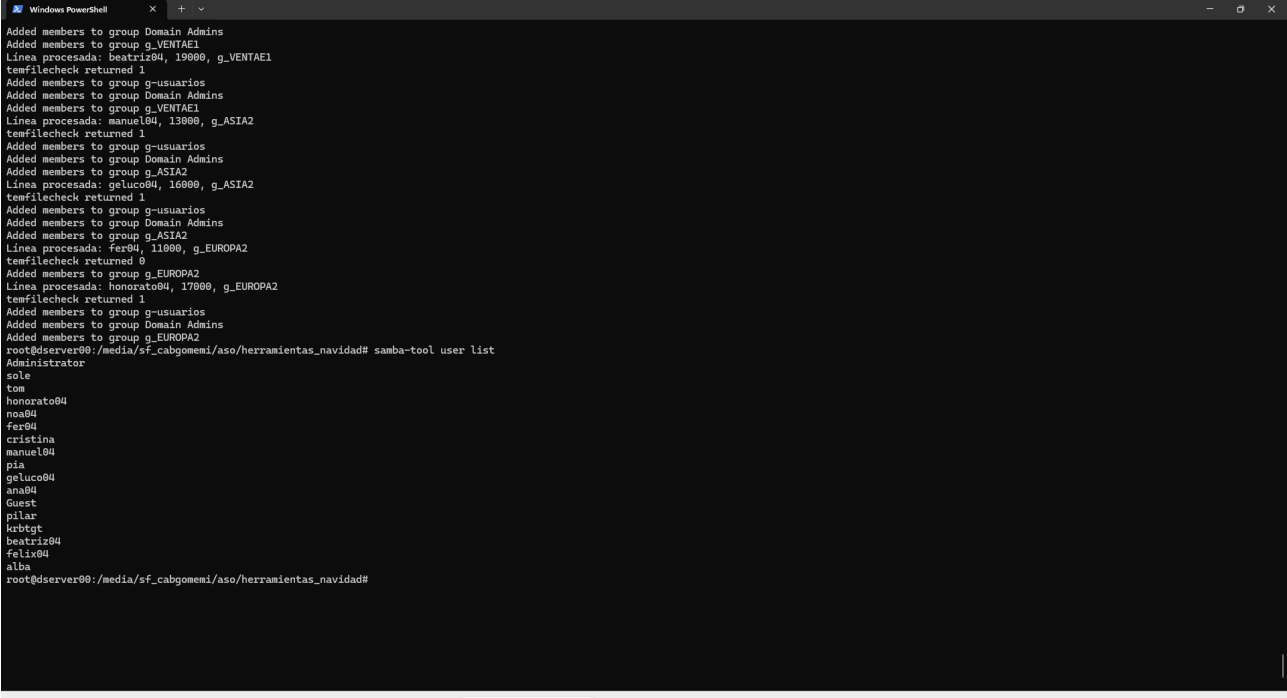
los añade mediante LDB

#Importacion de los grupos

```
ldbadd --url=/var/lib/samba/private/sam.ldb $OUTPUT_G
```

```
ldbadd --url=/var/lib/samba/private/sam.ldb $OUTPUT_U
```

comprobamos que se han añadido



```
Windows PowerShell
Added members to group Domain Admins
Added members to group g-VENTAE1
Linea procesada: beatriz04, 19000, g-VENTAE1
testfilecheck returned 1
Added members to group g-usuarios
Added members to group Domain Admins
Added members to group g-VENTAE1
Linea procesada: manuel04, 13000, g-ASIA2
testfilecheck returned 1
Added members to group g-usuarios
Added members to group Domain Admins
Added members to group g-ASIA2
Linea procesada: geluco04, 16000, g-ASIA2
testfilecheck returned 1
Added members to group g-usuarios
Added members to group Domain Admins
Added members to group g-ASIA2
Linea procesada: fer04, 11000, g-EUROPA2
testfilecheck returned 0
Added members to group g-EUROPA2
Linea procesada: honerato04, 17000, g-EUROPA2
testfilecheck returned 1
Added members to group g-usuarios
Added members to group Domain Admins
Added members to group g-EUROPA2
root@server00:/media/sf_cabgomemi/aso/herramientas_navidad# samba-tool user list
Administrator
sole
tom
honerato04
noa04
fer04
cristina
manuel04
pia
geluco04
ana04
Guest
pilar
krbtgt
beatriz04
felix04
alba
root@server00:/media/sf_cabgomemi/aso/herramientas_navidad#
```

El empleo de funciones en este script es esencial para evitar la longitud inadecuada y la reusabilidad de funcionalidades.

5.1 UTILS MANAGER

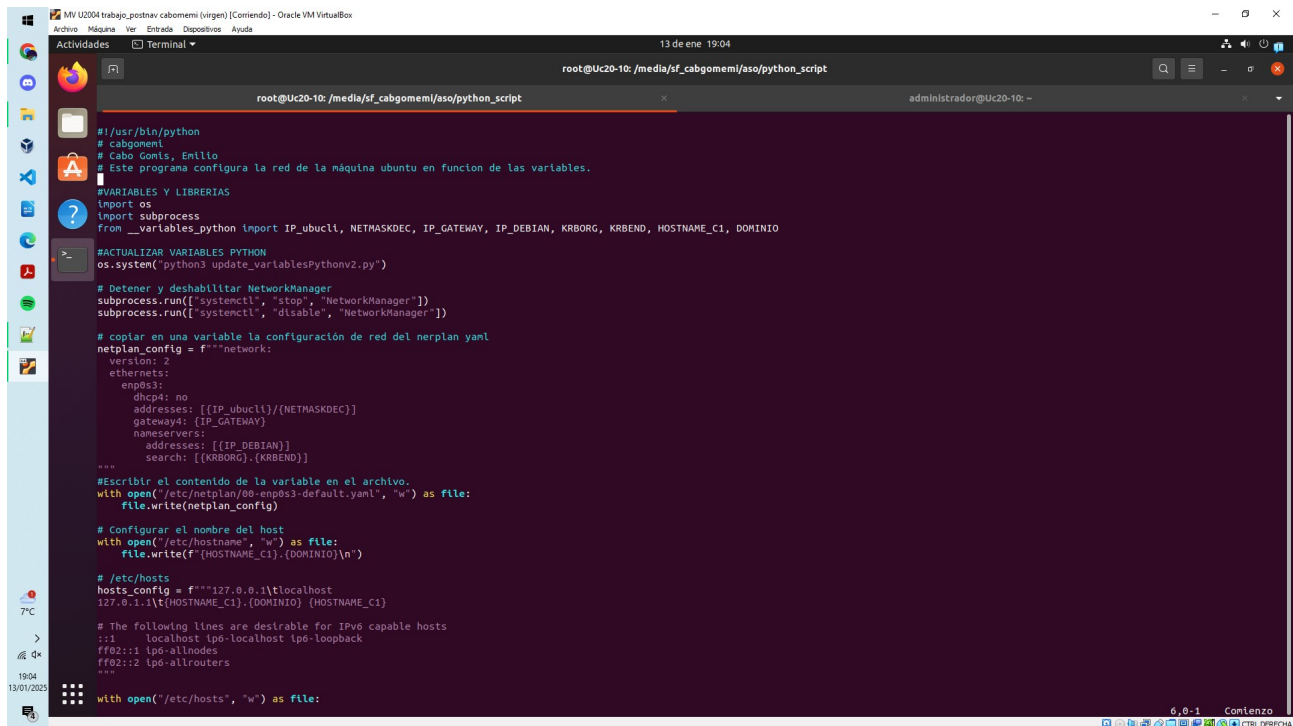
Python

En este script generaré los usuarios locales de manera automática.

Primero introduce la configuración del yaml en una variable para tener bien las tabulaciones.

Para ello empleamos “”” tripe comilla doble lo cual permite guardar el texto tal cual.

Debido a que las tabulaciones en python son parte del lenguaje tendremos que empezar con el contenido del archivo en la misma linea que declaramos y poblamos la variable



```
#!/usr/bin/python
# cabgomeml
# Cabo Gomis, Emilio
# Este programa configura la red de la máquina ubuntu en funcion de las variables.

# VARIABLES Y LIBRERIAS
import os
import subprocess
from __variables_python import IP_ubucll, NETMASKDEC, IP_GATEWAY, IP_DEBIAN, KRBORG, KRBEND, HOSTNAME_C1, DOMINIO

#ACTUALIZAR VARIABLES PYTHON
os.system("python3 update_variablesPythonv2.py")

# Detener y deshabilitar NetworkManager
subprocess.run(["systemctl", "stop", "NetworkManager"])
subprocess.run(["systemctl", "disable", "NetworkManager"])

# copiar en una variable la configuración de red del netplan yaml
netplan_config = """network:
  version: 2
  ethernet:
    enp8s3:
      dhcp4: no
      addresses: [[IP_ubucll]](NETMASKDEC)
      gateway4: [IP_GATEWAY]
      nameservers:
        addresses: [[IP_DEBIAN]]
        search: [[KRBORG],[KRBEND]]
"""

#Escribir el contenido de la variable en el archivo.
with open("/etc/netplan/00-enp8s3-default.yaml", "w") as file:
    file.write(netplan_config)

# Configurar el nombre del host
with open("/etc/hostname", "w") as file:
    file.write(f"{HOSTNAME_C1}.{DOMINIO}\n")

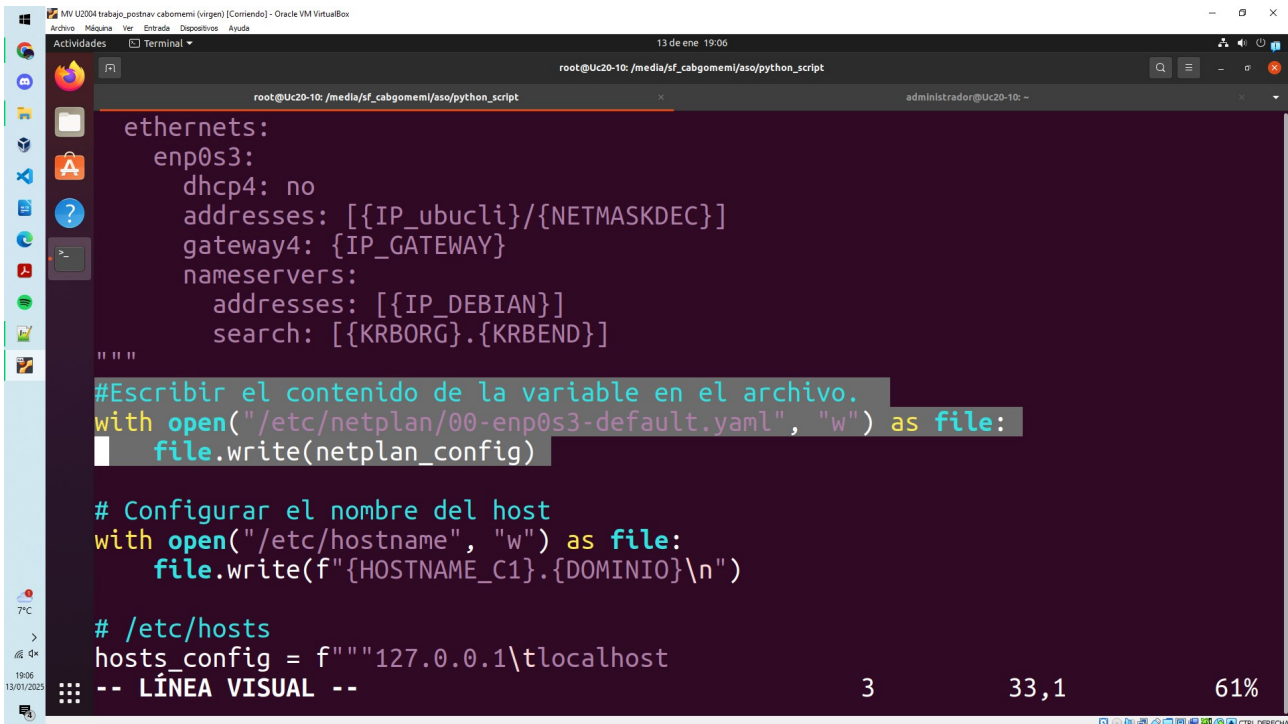
# /etc/hosts
hosts_config = """127.0.0.1localhost
127.0.0.1::1[HOSTNAME_C1].[DOMINIO] [HOSTNAME_C1]

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
"""

with open("/etc/hosts", "w") as file:
```

Luego abrimos el archivo para escritura.

5.1 UTILS MANAGER



```
ethernets:
  enp0s3:
    dhcp4: no
    addresses: [{IP_ubucli}/{NETMASKDEC}]
    gateway4: {IP_GATEWAY}
    nameservers:
      addresses: [{IP_DEBIAN}]
      search: [{KRBORG}.{KRBEND}]
"""

#Escribir el contenido de la variable en el archivo.
with open("/etc/netplan/00-enp0s3-default.yaml", "w") as file:
    file.write(netplan_config)

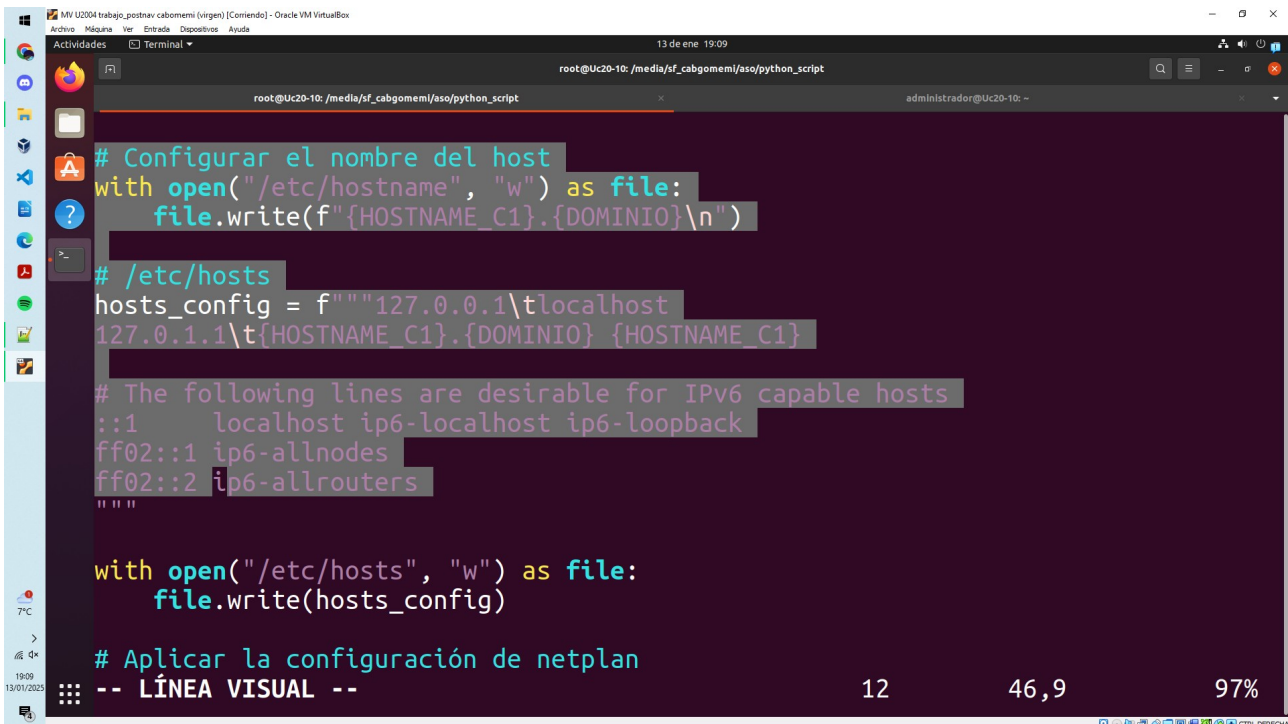
# Configurar el nombre del host
with open("/etc/hostname", "w") as file:
    file.write(f"{HOSTNAME_C1}.{DOMINIO}\n")

# /etc/hosts
hosts_config = f"""127.0.0.1\tlocalhost
::: -- LÍNEA VISUAL --
```

3 33,1 61%

OJO: si en lugar de `with open()` empleamos `open()` tendremos que luego aplicarle la función `close()` para que los contenidos del archivo se guarden correctamente.

Hacemos lo mismo con el nombre de host



```
# Configurar el nombre del host
with open("/etc/hostname", "w") as file:
    file.write(f"{HOSTNAME_C1}.{DOMINIO}\n")

# /etc/hosts
hosts_config = f"""127.0.0.1\tlocalhost
127.0.1.1\t{HOSTNAME_C1}.{DOMINIO} {HOSTNAME_C1}

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
"""

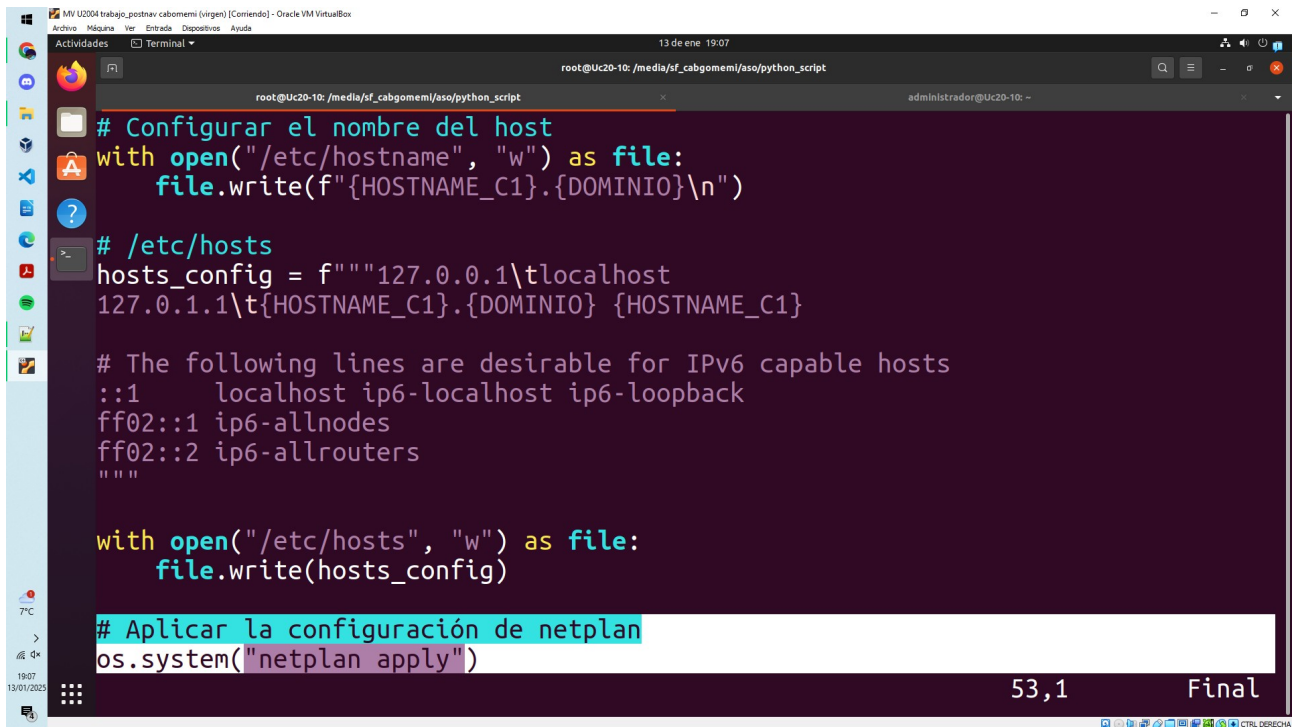
with open("/etc/hosts", "w") as file:
    file.write(hosts_config)

# Aplicar la configuración de netplan
::: -- LÍNEA VISUAL --
```

12 46,9 97%

5.1 UTILS MANAGER

Aplicamos la configuración. Este es un comando que al aplicarlo con la función `os.system()` lo interpreta BASH



```
root@Uc20-10: /media/sf_cabgomemi/aso/python_script
# Configurar el nombre del host
with open("/etc/hostname", "w") as file:
    file.write(f"{HOSTNAME_C1}.{DOMINIO}\n")

# /etc/hosts
hosts_config = f"""127.0.0.1\tlocalhost
127.0.1.1\t{HOSTNAME_C1}.{DOMINIO} {HOSTNAME_C1}

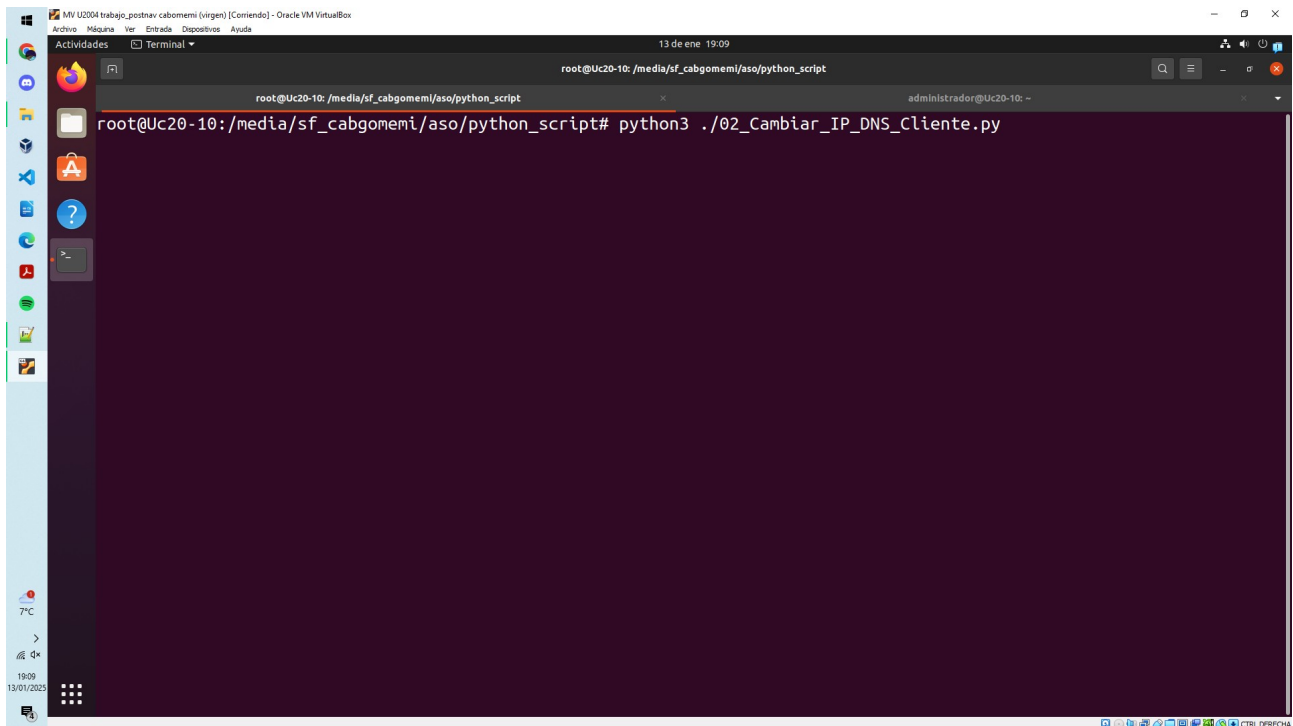
# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
"""

with open("/etc/hosts", "w") as file:
    file.write(hosts_config)

# Aplicar la configuración de netplan
os.system("netplan apply")
```

53,1 Final

Ejecutamos el comando.



```
root@Uc20-10: /media/sf_cabgomemi/aso/python_script
root@Uc20-10: /media/sf_cabgomemi/aso/python_script# python3 ./02_Cambiar_IP_DNS_Cliente.py
```

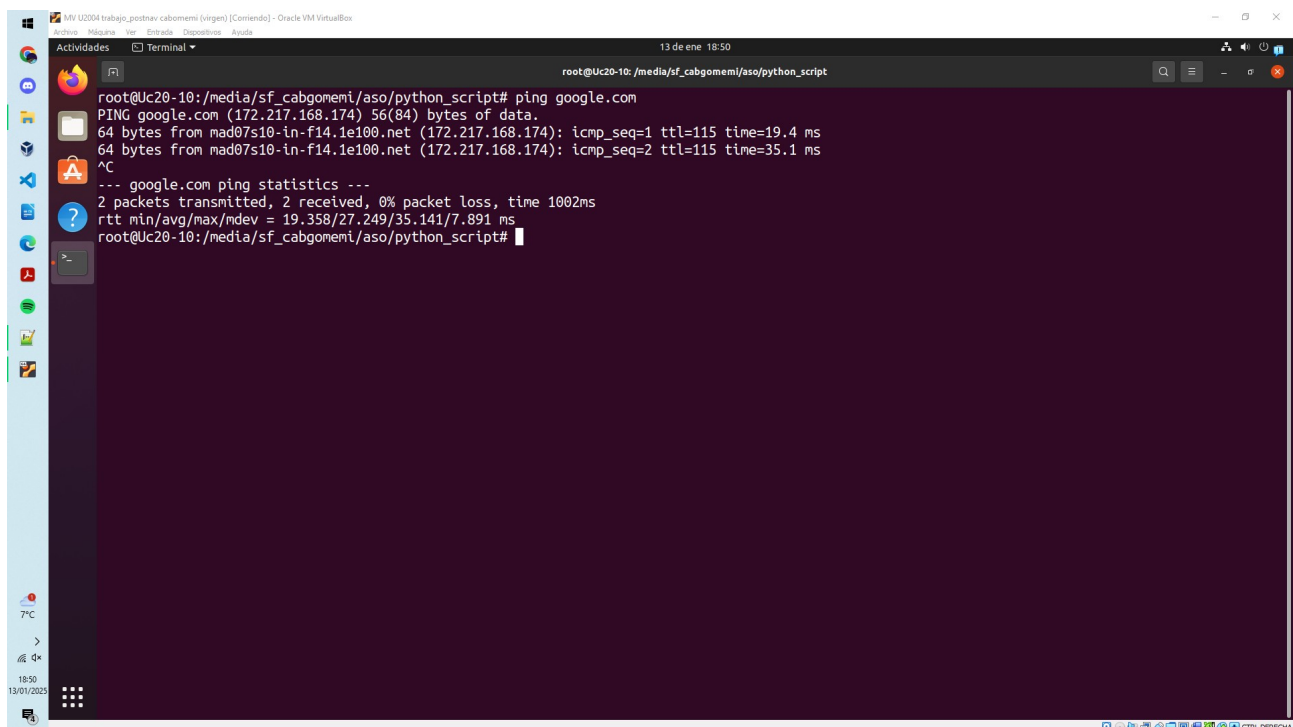
Comprobaciones

Comprobamos que el contenido del archivo de configuración resultante es el correcto.

```
root@Uc20-10:/media/sf_cabgomemi/aso/python_script# cat /etc/netplan/00-enp0s3-default.yaml
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: no
      addresses: [172.30.2.22/24]
      gateway4: 172.30.2.1
      nameservers:
        addresses: [172.30.2.21]
        search: [lpyme10.net]
root@Uc20-10:/media/sf_cabgomemi/aso/python_script# cat /etc/hostname
Uc20-10.lpyme10.net
root@Uc20-10:/media/sf_cabgomemi/aso/python_script# cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      Uc20-10.lpyme10.net Uc20-10

# The following lines are desirable for IPv6 capable hosts
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
root@Uc20-10:/media/sf_cabgomemi/aso/python_script#
```

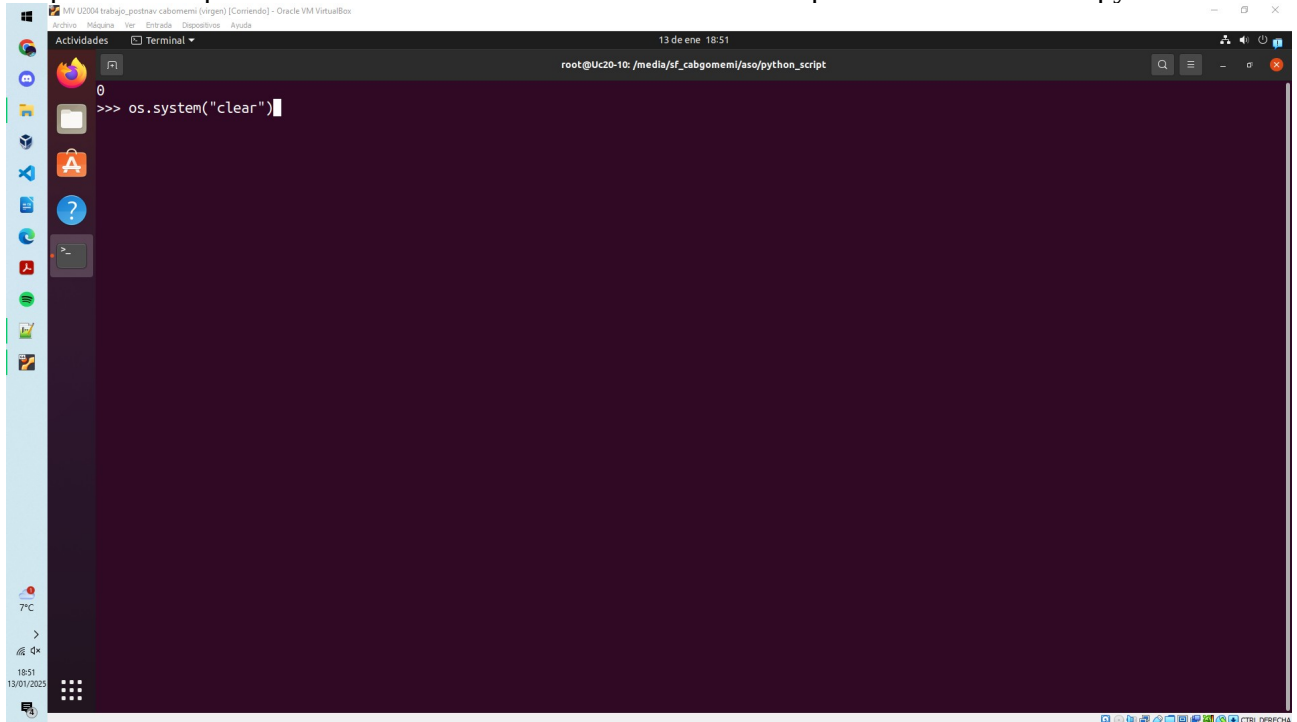
Comprobamos que la máquina puede conectarse a internet tras realizar la configuración de red.

A screenshot of a terminal window within an Oracle VM VirtualBox environment. The terminal shows the execution of a ping command to google.com. The output indicates that the connection is successful, with 2 packets transmitted, 2 received, and 0% packet loss. The time taken for the ping is 1002ms. The terminal window is titled 'root@Uc20-10: /media/sf_cabgomemi/aso/python_script' and is running on a system named 'Uc20-10'. The background of the terminal is dark purple.

```
root@Uc20-10:/media/sf_cabgomemi/aso/python_script# ping google.com
PING google.com (172.217.168.174) 56(84) bytes of data:
64 bytes from mad07s10-in-f14.1e100.net (172.217.168.174): icmp_seq=1 ttl=115 time=19.4 ms
64 bytes from mad07s10-in-f14.1e100.net (172.217.168.174): icmp_seq=2 ttl=115 time=35.1 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 19.358/27.249/35.141/7.891 ms
root@Uc20-10:/media/sf_cabgomemi/aso/python_script#
```

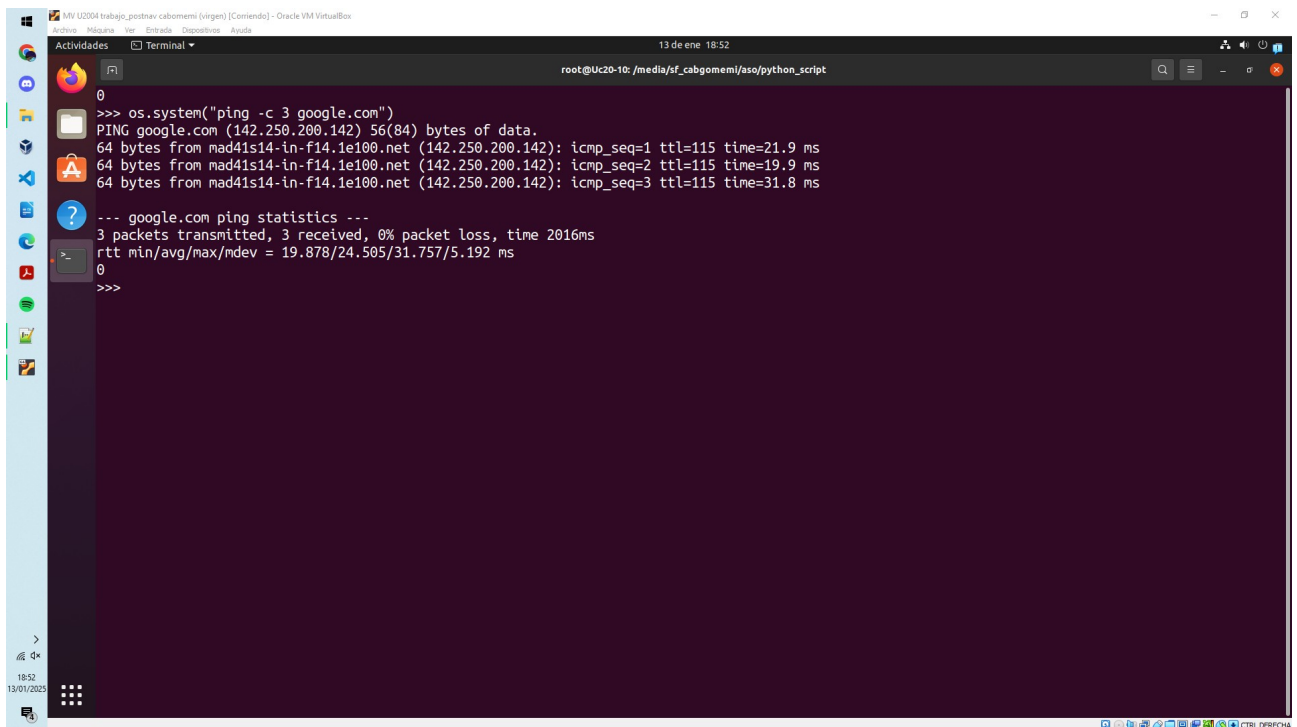
5.1 UTILS MANAGER

Si quisiéramos podríamos correr estos comandos desde el intérprete de comandos de python3



A screenshot of a terminal window within an Oracle VM VirtualBox. The window title is "root@Uc20-10: /media/sf_cabgomem/aso/python_script". The terminal shows a Python prompt `>>>` followed by the command `os.system("clear")`. The terminal background is dark purple. The host OS desktop is visible on the left, showing a taskbar with various application icons and a system tray at the bottom with a date and time display.

```
root@Uc20-10: /media/sf_cabgomem/aso/python_script
>>> os.system("clear")
```



A screenshot of a terminal window within an Oracle VM VirtualBox, showing the output of a ping command executed via Python. The window title is "root@Uc20-10: /media/sf_cabgomem/aso/python_script". The terminal shows a Python prompt `>>>` followed by the command `os.system("ping -c 3 google.com")`. The output displays three successful ping attempts to google.com (142.250.200.142) with varying response times. The terminal background is dark purple. The host OS desktop is visible on the left, showing a taskbar with various application icons and a system tray at the bottom with a date and time display.

```
root@Uc20-10: /media/sf_cabgomem/aso/python_script
>>> os.system("ping -c 3 google.com")
PING google.com (142.250.200.142) 56(84) bytes of data.
64 bytes from mad41s14-in-f14.1e100.net (142.250.200.142): icmp_seq=1 ttl=115 time=21.9 ms
64 bytes from mad41s14-in-f14.1e100.net (142.250.200.142): icmp_seq=2 ttl=115 time=19.9 ms
64 bytes from mad41s14-in-f14.1e100.net (142.250.200.142): icmp_seq=3 ttl=115 time=31.8 ms

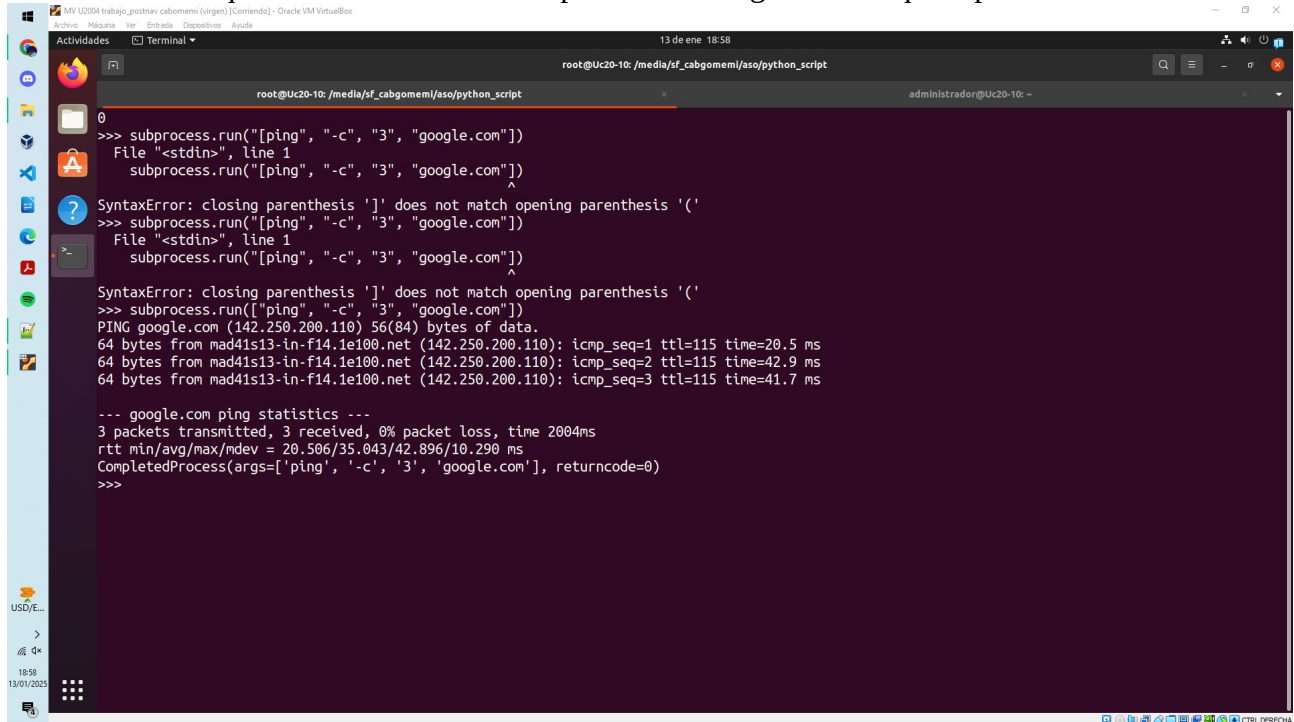
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2016ms
rtt min/avg/max/mdev = 19.878/24.505/31.757/5.192 ms
>>>
```

5.1 UTILS MANAGER

Otra forma de correr el comando de comprobación es el subprocess.run de la biblioteca subprocess.

Import subprocess

La diferencia es que mediante este método podemos configurar la máquina para



```
root@Uc20-10: /media/sf_cabgomeml/aso/python_script
0
>>> subprocess.run(["ping", "-c", "3", "google.com"])
File "<stdin>", line 1
  subprocess.run(["ping", "-c", "3", "google.com"])
      ^
SyntaxError: closing parenthesis ']' does not match opening parenthesis '('
>>> subprocess.run(["ping", "-c", "3", "google.com"])
File "<stdin>", line 1
  subprocess.run(["ping", "-c", "3", "google.com"])
      ^
SyntaxError: closing parenthesis ']' does not match opening parenthesis '('
>>> subprocess.run(["ping", "-c", "3", "google.com"])
PING google.com (142.250.200.110) 56(84) bytes of data.
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=1 ttl=115 time=20.5 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=2 ttl=115 time=42.9 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=3 ttl=115 time=41.7 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 20.506/35.043/42.896/10.290 ms
CompletedProcess(args=['ping', '-c', '3', 'google.com'], returncode=0)
>>>
```