

ЗЕРТХАНАЛЫҚ ЖҰМЫСТЫ орындауға арналған ӘДІСТЕМЕЛІК НҰСҚАУЛЫҚ

Huawei eNSP симуляторын қолданған жағдайда

Тақырыбы: Network Automation. Python

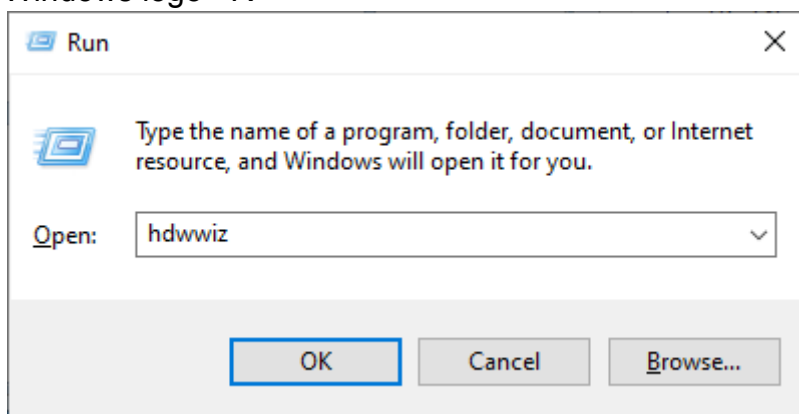
Жұмыстың мақсаты: HCIA-Datacom деңгейін меңгеру

Жұмыстың орындалу қадамы:

- 1) Windows операциялық жүйесінде Loopback interface құру
- 2) Windows операциялық жүйесімен eNSP симуляторын байланыстыру
- 3) Ubuntu Linux дистрибутивімен eNSP симуляторын байланыстыру
- 4) Telnet конфигурациялау
- 5) [Python TELNET Library](#)
- 6) SSH конфигурациялау
- 7) [Python Netmiko Library](#)
- 8) [Python NAPALM Library](#)

1-қадам: Windows операциялық жүйесінде Loopback interface құру

Windows logo +R



install the hardware ...

Add Hardware

The wizard can help you install other hardware

The wizard can search for other hardware and automatically install it for you. Or, if you know exactly which hardware model you want to install, you can select it from a list.

What do you want the wizard to do?

☐ Search for and install the hardware automatically (Recommended)

☒ Install the hardware that I manually select from a list (Advanced)

< Back Next > Cancel










Network adapters

Add Hardware

From the list below, select the type of hardware you are installing

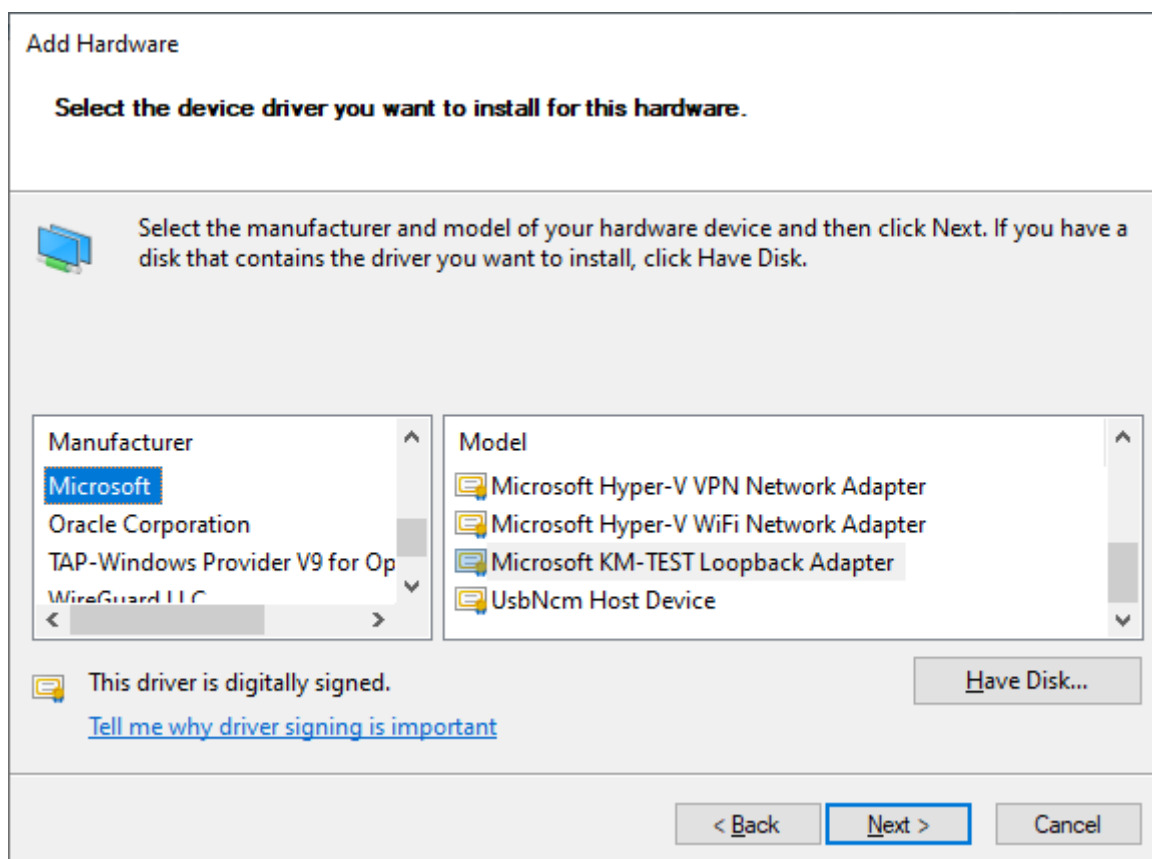
If you do not see the hardware category you want, click Show All Devices.

Common hardware types:

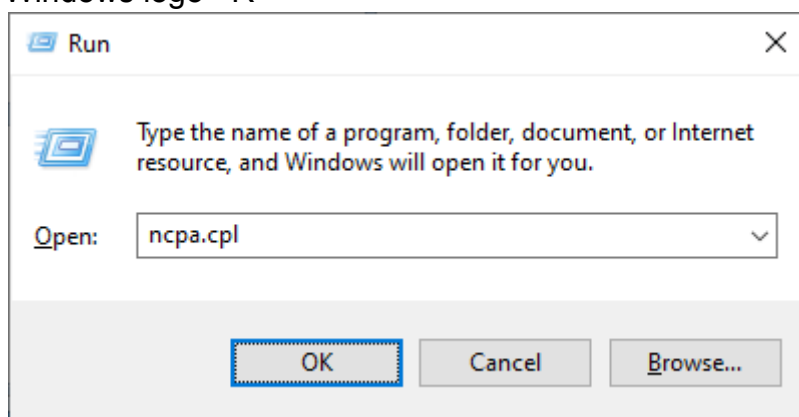
-  Modems
-  Multi-port serial adapters
-  Network adapters
-  OPOS Legacy Device
-  PCMCIA adapters
-  Perception Simulation Controllers
-  Portable Devices
-  Ports (COM & LPT)
-  POS Barcode Scanner

< Back Next > Cancel

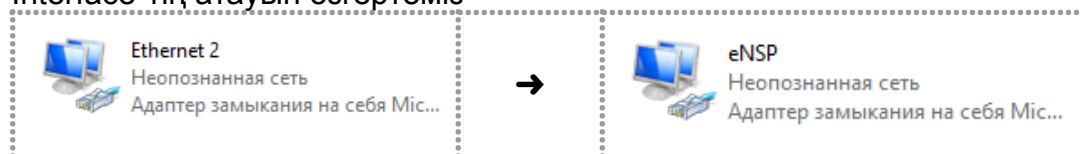
Microsoft → Microsoft KM-TEST Loopback Adapter



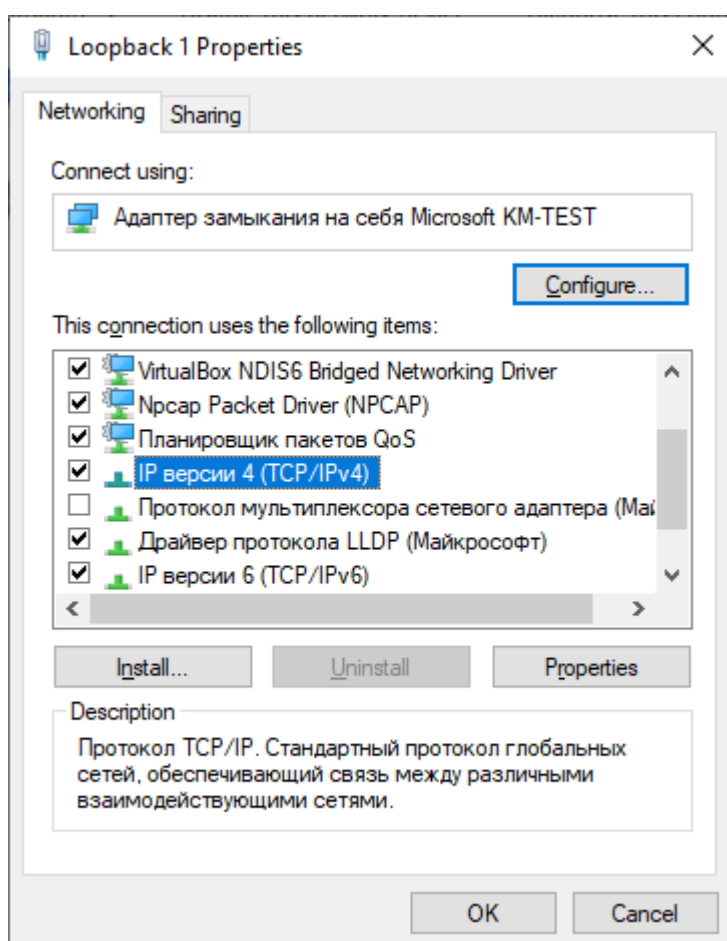
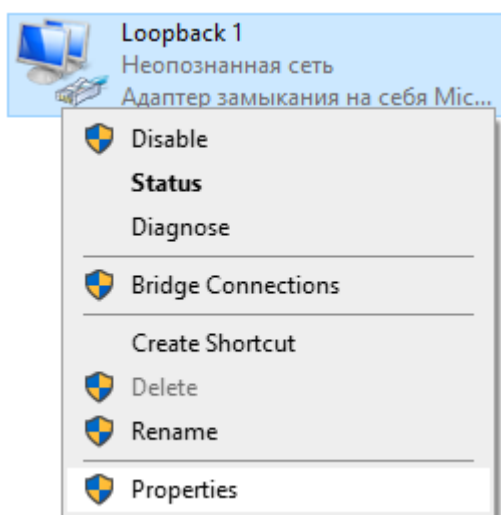
Windows logo +R

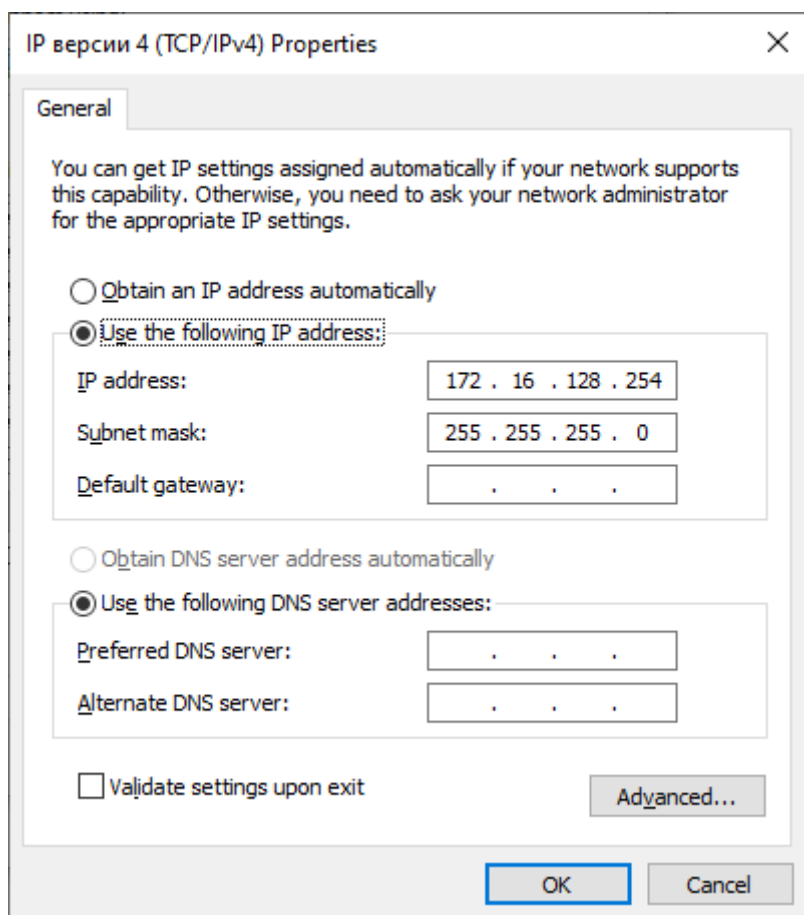


Interface-тің атауын өзгертеміз



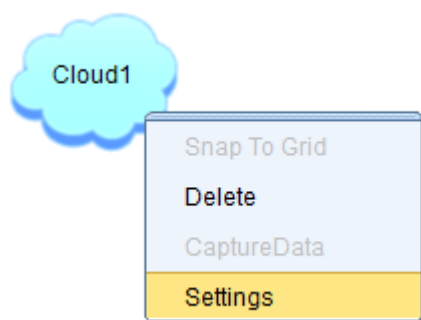
Loopback interface → Properties → TCP/IPv4 → IP address енгізу





PC/Laptop → Reboot

2-қадам: Windows операциялық жүйесімен eNSP симуляторын байланыстыру



Port Building → BindingInfo: UDP
 → Port Type: GE
 → “Add” батырманы басу

Cloud1

IO Config

Port Building

BindingInfo: UDP

Warning: Please don't bind the public network, otherwise maybe breakdown.

Port Type: GE ☐ Public UDP Port

Listening Port: 30000

Suggestion: (30000-35000)

Peer IP: 0 . 0 . 0 . 0

Peer Port: 0

Modify

Add Delete

No.	Port Type	Port Num	UDP Port	Port Open Status	Binding Info
1	GE	1	4666	Internal	UDP

Port Map Setting

Port Type: Ethernet

Local Port Num:

Remote Port Num:

☐ Two-way Channel

Add

Port Mapping

No.	Local Port Num	Remote Port Num	Port Type

Delete

Port Building:

→ BindingInfo: eNSP

→ Port Type: GE

→ “Add” батырманы басу

Cloud1

IO Config

Port Building

BindingInfo: eNSP -- IP: 172.16.128.254

Warning: Please don't bind the public network, otherwise maybe breakdown.

Port Type: GE ☐ Public UDP Port

Listening Port: 30000

Suggestion: (30000-35000)

Peer IP: 0 . 0 . 0 . 0

Peer Port: 0

No.	Port Type	Port Num	UDP Port	Port Open Status	Binding Info
1	GE	1	4666	Internal	UDP
2	GE	2	None	Public	eNSP -- IP: 172.16.128.254

Port Map Setting

Port Type: Ethernet

Local Port Num:

Remote Port Num:

☐ Two-way Channel

Port Mapping

No.	Local Port Num	Remote Port Num	Port Type

Port Map Setting:

→ Port Type: GE

→ Local Port Num: 1

→ Remote Port Num: 2

☒ Two-way Channel

→ “Add” батырманы басу

Cloud1

IO Config

Port Building

BindingInfo: eNSP -- IP: 172.16.128.254
 Warning: Please don't bind the public network, otherwise maybe breakdown.

Port Type: GE ☐ Public UDP Port

Listening Port: 30000
 Suggestion: (30000-35000)

Peer IP: 0 . 0 . 0 . 0
 Peer Port: 0

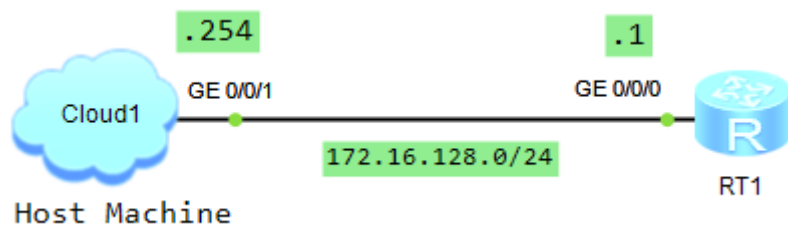
No.	Port Type	Port Num	UDP Port	Port Open Status	Binding Info
1	GE	1	4666	Internal	UDP
2	GE	2	None	Public	eNSP -- IP: 172.16.128.254

Port Map Setting

Port Type: GE
 Local Port Num: 1
 Remote Port Num: 2
☒ Two-way Channel

Port Mapping

No.	Local Port Num	Remote Port Num	Port Type
1	1	2	GE
2	2	1	GE



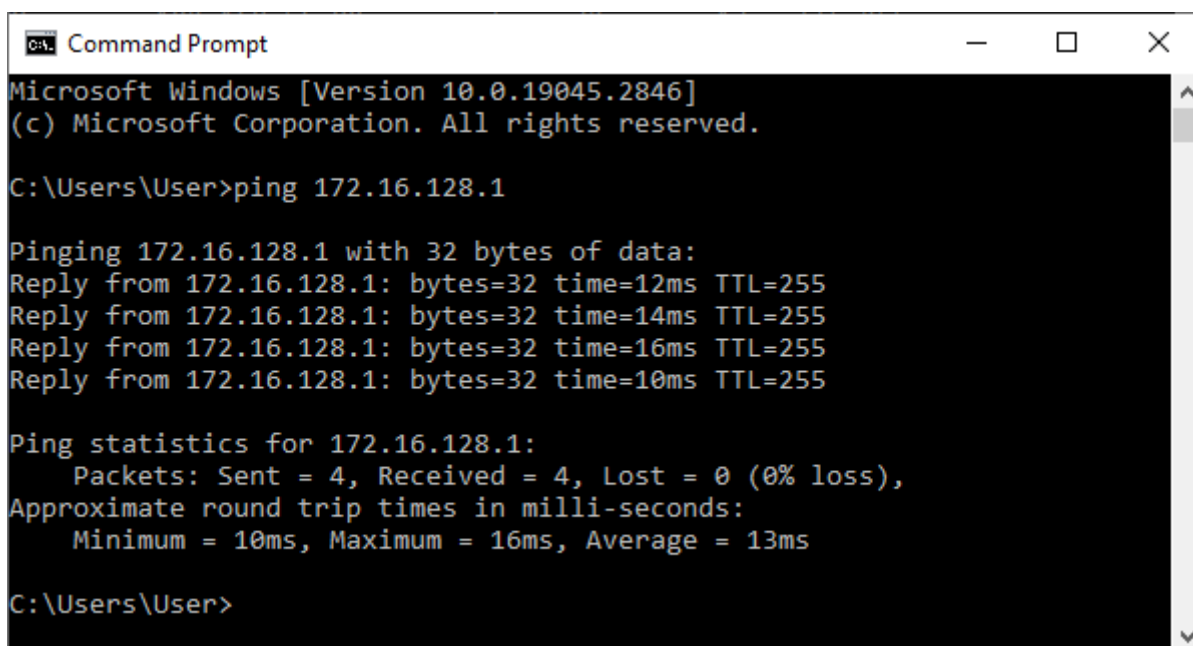
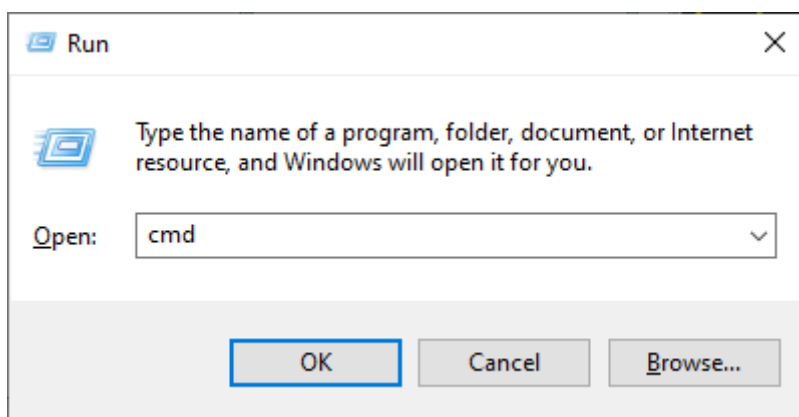
```
<Huawei> undo terminal logging
<Huawei> system-view
[Huawei] sysname RT1

[RT1] interface g0/0/0
[RT1-GigabitEthernet0/0/0] ip address 172.16.128.1 24

[RT1] display ip int brief
```

Interface	IP Address/Mask	Physical	Protocol
GigabitEthernet0/0/0	172.16.128.1/24	up	up
GigabitEthernet0/0/1	unassigned	down	down
GigabitEthernet0/0/2	unassigned	down	down
NULL0	unassigned	up	up(s)

Windows logo +R



3-қадам: Ubuntu Linux дистрибутивімен eNSP симуляторын байланыстыру

Download Linux images (OVF Files)

<https://drive.google.com/drive/folders/1cUMzXEw4BFvp3dGhMJQiGjQnVS0b3MTB?usp=sharing>

Download VMware Workstation Pro

<https://drive.google.com/drive/folders/1xPeOKfdeOzGdEHJRhYktJThgL6-xjkHy?usp=sharing>

Link YouTube Channel

<https://www.youtube.com/playlist?list=PLqzHIBlxTu2wlO2GhGKEV5edIE0z7SSb>

...

Ubuntu Linux жүйесіне кіру

login: student

Password: 123

student@ubuntu22-04~\$ ip address

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group de
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
link/ether 00:0c:29:a0:a0:d2 brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 192.168.0.104/24 metric 100 brd 192.168.0.255 scope global dynamic
    valid_lft 7171sec preferred_lft 7171sec
inet6 fe80::20c:29ff:fea0:a0d2/64 scope link
    valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
link/ether 00:0c:29:a0:a0:dc brd ff:ff:ff:ff:ff:ff
altname enp2s5

```

student@ubuntu22-04~\$ sudo nano /etc/netplan/00-installer-config.yaml

```

network:
  ethernets:
    ens33:
      dhcp4: true
    ens37:
      addresses: [172.16.128.5/24]
  version: 2

```

Ctrl+O → Enter

Ctrl+X

student@ubuntu22-04~\$ sudo netplan apply

student@ubuntu22-04~\$ ip address

```

2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
link/ether 00:0c:29:a0:a0:d2 brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 192.168.0.104/24 metric 100 brd 192.168.0.255 scope global dynamic
    valid_lft 5393sec preferred_lft 5393sec
inet6 fe80::20c:29ff:fea0:a0d2/64 scope link
    valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
link/ether 00:0c:29:a0:a0:dc brd ff:ff:ff:ff:ff:ff
altname enp2s5
inet 172.16.128.5/24 brd 172.16.128.255 scope global ens37
    valid_lft forever preferred_lft forever
inet6 fe80::20c:29ff:fea0:a0dc/64 scope link
    valid_lft forever preferred_lft forever

```

Ping from Linux to Windows

student@ubuntu22-04~\$ ping -c 4 172.16.128.254

```

64 bytes from 172.16.128.254: icmp_seq=1 ttl=128 time=0.453 ms
64 bytes from 172.16.128.254: icmp_seq=2 ttl=128 time=0.803 ms
64 bytes from 172.16.128.254: icmp_seq=3 ttl=128 time=0.466 ms
64 bytes from 172.16.128.254: icmp_seq=4 ttl=128 time=0.494 ms

```

Ping from Linux to Router

student@ubuntu22-04~\$ ping -c 4 172.16.128.1

```
64 bytes from 172.16.128.1: icmp_seq=1 ttl=255 time=11.1 ms
64 bytes from 172.16.128.1: icmp_seq=2 ttl=255 time=7.69 ms
64 bytes from 172.16.128.1: icmp_seq=3 ttl=255 time=13.0 ms
64 bytes from 172.16.128.1: icmp_seq=4 ttl=255 time=9.76 ms
```

4-қадам: Telnet конфигурациялау (Configure Telnet Remote Management)

```
[RT1] display telnet server status
```

```
TELNET IPV4 server      :Enable
TELNET IPV6 server      :Enable
TELNET server port      :23
```

```
[RT1] user-interface vty 0 4
```

```
[RT1-ui-vty0-4] authentication-mode aaa
```

```
[RT1] aaa
```

```
[RT1-aaa] local-user user1 password cipher user@123
```

```
[RT1-aaa] local-user user1 privilege level 15
```

```
[RT1-aaa] local-user user1 service-type telnet
```



```
student@ubuntu22-04~$ telnet
```

```
telnet> open 172.16.128.1
Trying 172.16.128.1...
Connected to 172.16.128.1.
Escape character is '^]'.

Login authentication

Username:user1
Password:
<RT1>system-view
Enter system view, return user view with Ctrl+Z.
[RT1]
```

5-қадам: Python TELNET Library

```
student@ubuntu22-04~$ python3 --version
```

```
Python 3.10.6
```

```
student@ubuntu22-04~$ ls -l
```

```
-rw-rw-r-- 1 student student 0 Apr 23 04:21 script1_telnet.py
-rw-rw-r-- 1 student student 0 Apr 23 04:21 script2_telnet.py
-rw-rw-r-- 1 student student 0 Apr 23 04:22 script3_telnet.py
```

student@ubuntu22-04~\$ nano script1_telnet.py

```
import telnetlib
import time

host = '172.16.128.1'
user = 'user1'
password = 'user@123'
UserPrompt = '>'
ConfigPrompt = ']'

tn = telnetlib.Telnet(host)

tn.read_until(b'Username:')
tn.write(user.encode('ascii') + b'\n')
tn.read_until(b'Password:')
tn.write(password.encode('ascii') + b'\n')

print('Connection to ' + '172.16.128.1' + ' is Successful')

UserMode = tn.read_until(UserPrompt.encode('ascii'))
print(UserMode.decode('ascii'))

tn.write(b'system-view \n')
ConfigMode = tn.read_until(ConfigPrompt.encode('ascii'))
print(ConfigMode.decode('ascii'))

tn.close()
Ctrl+O → Enter
Ctrl+X
```

student@ubuntu22-04~\$ python3 script1_telnet.py

```
Connection to 172.16.128.1 is Successful

<RT1>
system-view
Enter system view, return user view with Ctrl+Z.
[RT1]
student@ubuntu22-04:~$
```

student@ubuntu22-04~\$ nano script2_telnet.py

```
import telnetlib
import time

host = '172.16.128.1'
user = 'user1'
password = 'user@123'
UserPrompt = '>'
ConfigPrompt = ']'

tn = telnetlib.Telnet(host)

tn.read_until(b'Username:')
tn.write(user.encode('ascii') + b'\n')
tn.read_until(b'Password:')
tn.write(password.encode('ascii') + b'\n')
```

```

print('Connection to ' + '172.16.128.1' + ' is Successful')

UserMode = tn.read_until(UserPrompt.encode('ascii'))
print(UserMode.decode('ascii'))

tn.write(b'system-view \n')
ConfigMode = tn.read_until(ConfigPrompt.encode('ascii'))
print(ConfigMode.decode('ascii'))

tn.write(b'interface LoopBack 0 \n')
ConfigMode = tn.read_until(ConfigPrompt.encode('ascii'))
print(ConfigMode.decode('ascii'))

tn.write(b'ip address 50.0.1.1 24 \n')
ConfigMode = tn.read_until(ConfigPrompt.encode('ascii'))
print(ConfigMode.decode('ascii'))

tn.write(b'display ip interface brief \n')
ConfigMode = tn.read_until(ConfigPrompt.encode('ascii'))
print(ConfigMode.decode('ascii'))

tn.close()
Ctrl+O → Enter
Ctrl+X

```

student@ubuntu22-04~\$ python3 script2_telnet.py

```

Connection to 172.16.128.1 is Successful
<RT1>
system-view
Enter system view, return user view with Ctrl+Z.
[RT1]
interface LoopBack 0
[RT1-LoopBack0]
ip address 50.0.1.1 24
[RT1-LoopBack0]
display ip interface brief

Interface                IP Address/Mask      Physical  Protocol
GigabitEthernet0/0/0     172.16.128.1/24      up        up
GigabitEthernet0/0/1     unassigned            down      down
GigabitEthernet0/0/2     unassigned            down      down
LoopBack0                50.0.1.1/24          up        up(s)
NULL0                    unassigned            up        up(s)
[RT1-LoopBack0]
student@ubuntu22-04:~$

```

6-қадам: SSH конфигурациялау (Configure SSH Remote Management)

```

[RT1] rsa local-key-pair create
Confirm to replace them? (y/n)[n]: y
Input the bits in the modulus[default = 512]: 2048

[RT1] user-interface vty 0 4
[RT1-ui-vty0-4] authentication-mode aaa
[RT1-ui-vty0-4] protocol inbound all
[RT1-ui-vty0-4] display this

[RT1] aaa

```

```
[RT1-aaa] local-user user2 password cipher user@123
[RT1-aaa] local-user user2 privilege level 15
[RT1-aaa] local-user user2 service-type ssh
```

```
[RT1] stelnet server enable
```

```
Succeeded in starting the STELNET server
```

```
<RT1> display ssh server status
```

```
SSH version           :1.99
SSH connection timeout :60 seconds
SSH server key generating interval :0 hours
SSH Authentication retries :3 times
SFTP Server           :Disable
Stelnet server        :Enable
```

Қашықтан басқару (Remote Access)

```
student@ubuntu22-04~$ systemctl status ssh
```

```
Active: active (running)
```

```
student@ubuntu22-04~$ sudo nano .ssh/config
```

```
Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
```

```
KexAlgorithms +diffie-hellman-group-exchange-sha1,diffie-hellman-group1-sha1
```

```
HostKeyAlgorithms+=ssh-rsa
```

```
Ctrl+O → Enter
```

```
Ctrl+X
```

```
student@ubuntu22-04~$ ssh user2@172.16.128.1
```

```
The authenticity of host '172.16.128.1 (172.16.128.1)' can't be established.
RSA key fingerprint is SHA256:RP/9qVLMXrexBrzTCrnOTKtxFuh36ZOjo/6NdUhy/D8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.128.1' (RSA) to the list of known hosts.
user2@172.16.128.1's password:

<RT1>system-view
Enter system view, return user view with Ctrl+Z.
[RT1]
```

7-қадам: Python NetMiko Library

Paramiko – SSH2 protocol library

<https://pypi.org/project/paramiko/>

Netmiko – Multi-vendor library to simplify CLI connections to network devices

<https://pypi.org/project/netmiko/>

```
student@ubuntu22-04~$ ping -c4 google.com
```

```
64 bytes from 216.58.210.142: icmp_seq=1 ttl=111 time=69.0 ms
64 bytes from 216.58.210.142: icmp_seq=2 ttl=111 time=69.0 ms
64 bytes from 216.58.210.142: icmp_seq=3 ttl=111 time=69.2 ms
64 bytes from 216.58.210.142: icmp_seq=4 ttl=111 time=68.9 ms
```

```
student@ubuntu22-04~$ sudo apt update
student@ubuntu22-04~$ sudo apt install -y build-essential libssl-dev
libffi-dev
```

```
student@ubuntu22-04~$ python3 --version
```

```
Python 3.10.6
```

```
student@ubuntu22-04~$ sudo apt install -y python3-pip
```

```
student@ubuntu22-04~$ pip3 --version
```

```
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

```
student@ubuntu22-04~$ python3 -m venv netmiko_vrp
```

```
student@ubuntu22-04~$ source netmiko_vrp/bin/activate
```

```
(netmiko_vrp) student@ubuntu22-04~$ deactivate
```

```
student@ubuntu22-04~$ source netmiko_vrp/bin/activate
```

```
(netmiko_vrp) student@ubuntu22-04~$ python -m pip install paramiko
```

```
(netmiko_vrp) student@ubuntu22-04~$ python -m pip install netmiko
```

```
(netmiko_vrp) student@ubuntu22-04~$ python -m pip list
```

Package	Version
bcrypt	4.0.1
cffi	1.15.1
cryptography	40.0.2
future	0.18.3
netmiko	4.1.2
ntc-templates	3.3.0
paramiko	3.1.0
pip	22.0.2
pycparser	2.21
PyNaCl	1.5.0
pyserial	3.5
PyYAML	6.0
scp	0.14.5
setuptools	59.6.0
six	1.16.0
tenacity	8.2.2
textfsm	1.1.2

```
student@ubuntu22-04~$ ls -l
```

```
-rw-rw-r-- 1 student student 276 Apr 23 04:24 script1_netmiko.py
```

```
student@ubuntu22-04~$ nano script1_netmiko.py
```

```
from netmiko import ConnectHandler
```

```
AR2220 = {
    'device_type': 'huawei',
    'host': '172.16.128.1',
    'username': 'user2',
    'password': 'user@123'
}
```

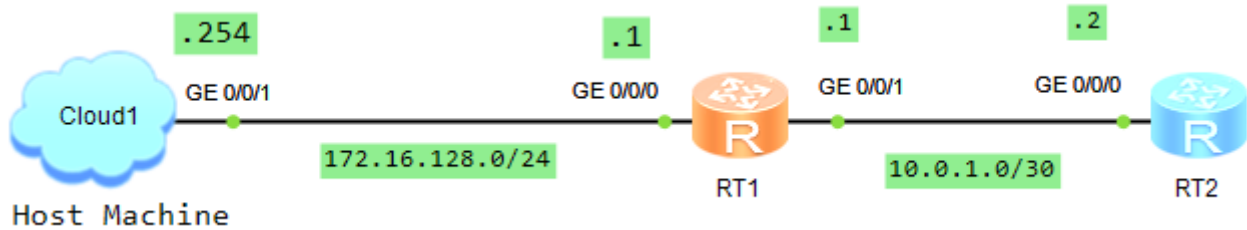
```
net_connect = ConnectHandler(**AR2220)
output = net_connect.send_command('display ip interface brief')
print(output)
```

Ctrl+O → Enter

Ctrl+X

(netmiko_vrp) student@ubuntu22-04~\$ python script1_netmiko.py

Interface	IP Address/Mask	Physical	Protocol
GigabitEthernet0/0/0	172.16.128.1/24	up	up
GigabitEthernet0/0/1	unassigned	down	down
GigabitEthernet0/0/2	unassigned	down	down
LoopBack0	50.0.1.1/24	up	up(s)
NULL0	unassigned	up	up(s)



student@ubuntu22-04~\$ nano script2_netmiko.py

```

from netmiko import ConnectHandler

AR2220 = {
    'device_type': 'huawei',
    'host': '172.16.128.1',
    'username': 'user2',
    'password': 'user@123'
}

net_connect = ConnectHandler(**AR2220)
output = net_connect.send_command('display current-configuration interface g0/0/1')
print(output)

commands = ['interface GigabitEthernet 0/0/1', 'ip address 10.0.1.1 30', 'display
this', 'ospf 1 router-id 1.1.1.1', 'area 0', 'network 10.0.1.0 0.0.0.3', network
172.16.128.0 0.0.0.255']
output = net_connect.send_config_set(commands)
print(output)

```

```

output = net_connect.send_command('display current-configuration section ospf')
print(output)

```

Ctrl+O → Enter

Ctrl+X

(netmiko_vrp) student@ubuntu22-04~\$ python script2_netmiko.py


```
#
interface GigabitEthernet0/0/1
#
return
system-view
Enter system view, return user view with Ctrl+Z.
[RT1]interface GigabitEthernet 0/0/1
[RT1-GigabitEthernet0/0/1]ip address 10.0.1.1 30
[RT1-GigabitEthernet0/0/1]display this
[V200R003C00]
#
interface GigabitEthernet0/0/1
 ip address 10.0.1.1 255.255.255.252
#
return
[RT1-GigabitEthernet0/0/1]ospf 1 router-id 1.1.1.1
[RT1-ospf-1]area 0
[RT1-ospf-1-area-0.0.0.0]network 10.0.1.0 0.0.0.3
[RT1-ospf-1-area-0.0.0.0]network 172.16.128.0 0.0.0.255
[RT1-ospf-1-area-0.0.0.0]return
<RT1>
[V200R003C00]
#
ospf 1 router-id 1.1.1.1
 area 0.0.0.0
  network 10.0.1.0 0.0.0.3
  network 172.16.128.0 0.0.0.255
#
return
```

8-қадам: Python NAPALM Library

NAPALM's documentation

<https://napalm.readthedocs.io>

NAPALM (Network Automation and Programmability Abstraction Layer with Multivendor support) is a Python library that implements a set of functions to interact with different network device Operating Systems using a unified API.

NAPALM Huawei VRP

<https://github.com/napalm-automation-community/napalm-huawei-vrp>

<https://pypi.org/project/napalm-huawei-vrp/>

```
student@ubuntu22-04~$ python3 -m venv napalm_vrp
```

```
student@ubuntu22-04~$ source napalm_vrp/bin/activate
```

```
(napalm_vrp) student@ubuntu22-04~$ python -m pip install napalm-huawei-vrp
```

```
(napalm_vrp) student@ubuntu22-04~$ python -m pip list
```

```
student@ubuntu22-04~$ ls -l
```

```
-rw-rw-r-- 1 student student 283 Apr 23 04:50 script1_napalm.py
-rw-rw-r-- 1 student student 294 Apr 23 04:53 script2_napalm.py
-rw-rw-r-- 1 student student 278 Apr 23 04:55 script3_napalm.py
-rw-rw-r-- 1 student student 288 Apr 23 04:56 script4_napalm.py
```

```
student@ubuntu22-04~$ nano script1_napalm.py
```

```
from napalm import get_network_driver
import pprint

driver = get_network_driver('huawei_vrp')
device = driver(hostname='172.16.128.1', username='user2', password='user@123')
device.open()

# Get Facts API. Return general device information
get_facts = device.get_facts()
pprint.pprint(get_facts)
Ctrl+O → Enter
Ctrl+X
```

```
(napalm_vrp) student@ubuntu22-04~$ python script1_napalm.py
```

```
{'fqdn': 'Unknown',
 'hostname': 'RT1',
 'interface_list': ['GigabitEthernet0/0/0',
                    'GigabitEthernet0/0/1',
                    'GigabitEthernet0/0/2',
                    'NULL0'],
 'model': 'Unknown',
 'os_version': 'Unknown',
 'serial_number': [],
 'uptime': -1,
 'vendor': 'Huawei'}
```

```
student@ubuntu22-04~$ nano script2_napalm.py
```

```
from napalm import get_network_driver
import pprint

driver = get_network_driver('huawei_vrp')
device = driver(hostname='172.16.128.1', username='user2', password='user@123')
device.open()

# Get Interfaces IP API
get_interfaces_ip = device.get_interfaces_ip()
pprint.pprint(get_interfaces_ip)
Ctrl+O → Enter
Ctrl+X
```

```
(napalm_vrp) student@ubuntu22-04~$ python script2_napalm.py
```

```
{'GigabitEthernet0/0/0': {'ipv4': {'172.16.128.1': {'prefix_length': 24}}}}
```

```
student@ubuntu22-04~$ nano script3_napalm.py
```

```
from napalm import get_network_driver
import pprint

driver = get_network_driver('huawei_vrp')
device = driver(hostname='172.16.128.1', username='user2', password='user@123')
device.open()

# Ping API
ping_remote_ip = device.ping('172.16.128.254')
pprint.pprint(ping_remote_ip)
```

Ctrl+O → Enter

Ctrl+X

student@ubuntu22-04~\$ python3 script3_napalm.py

```
{'success': {'packet_loss': 0,
             'probes_sent': 5,
             'results': [{'ip_address': '172.16.128.254', 'rtt': 10.0},
                          {'ip_address': '172.16.128.254', 'rtt': 10.0},
                          {'ip_address': '172.16.128.254', 'rtt': 10.0},
                          {'ip_address': '172.16.128.254', 'rtt': 1.0},
                          {'ip_address': '172.16.128.254', 'rtt': 10.0}],
             'rtt_avg': 8.0,
             'rtt_max': 10.0,
             'rtt_min': 1.0,
             'rtt_stddev': 0.0}}
```

student@ubuntu22-04~\$ nano script4_napalm.py

```
from napalm import get_network_driver
import pprint

driver = get_network_driver('huawei_vrp')
device = driver(hostname='172.16.128.1', username='user2', password='user@123')
device.open()
```

```
# Send Any CLI command
send_command = device.cli(['display version'])
pprint.pprint(send_command)
```

Ctrl+O → Enter

Ctrl+X

student@ubuntu22-04~\$ python3 script4_napalm.py

```
{'display version': 'Huawei Versatile Routing Platform Software\n'
                    'VRP (R) software, Version 5.130 (AR2200 V200R003C00)\n'
                    'Copyright (C) 2011-2012 HUAWEI TECH CO., LTD\n'
                    'Huawei AR2220 Router uptime is 0 week, 0 day, 0 hour, 58 '\n'
                    'minutes\n'
                    'BKP 0 version information: \n'
                    '1. PCB      Version   : AR01BAK2A VER.NC\n'
                    '2. If Supporting PoE : No\n'
                    '3. Board   Type      : AR2220\n'
                    '4. MPU Slot Quantity : 1\n'
                    '5. LPU Slot Quantity : 6\n'
                    '\n'
                    'MPU 0(Master) : uptime is 0 week, 0 day, 0 hour, 58 '\n'
                    'minutes\n'
                    'MPU version information : \n'
                    '1. PCB      Version   : AR01SRU2A VER.A\n'
                    '2. MAB      Version   : 0\n'
                    '3. Board   Type      : AR2220\n'
                    '4. BootROM  Version   : 0\n'}
```

student@ubuntu22-04~\$ ls -l

```
-rw-rw-r-- 1 student student 93 Apr 23 06:05 ospf_config
-rw-rw-r-- 1 student student 449 Apr 23 06:06 script5_napalm.py
```

student@ubuntu22-04~\$ nano ospf_config

```
ospf 1 router-id 1.1.1.1
area 0
interface g0/0/1
ip address 10.0.1.1 30
ospf enable 1 area 0
```

Ctrl+O → Enter

Ctrl+X

student@ubuntu22-04~\$ nano script5_napalm.py

```
from napalm import get_network_driver
import pprint

driver = get_network_driver('huawei_vrp')
device = driver(hostname='172.16.128.1', username='user2', password='user@123')
device.open()

command = ['display current-configuration interface g0/0/1']
get_interface = device.cli(command)
pprint.pprint(get_interface)

device.load_merge_candidate('ospf_config')
device.commit_config()

get_interface = device.cli(command)
pprint.pprint(get_interface)
Ctrl+O → Enter
Ctrl+X
```