
CNIT-381
FALL 2020

NETMIKO

Low Level Interactions

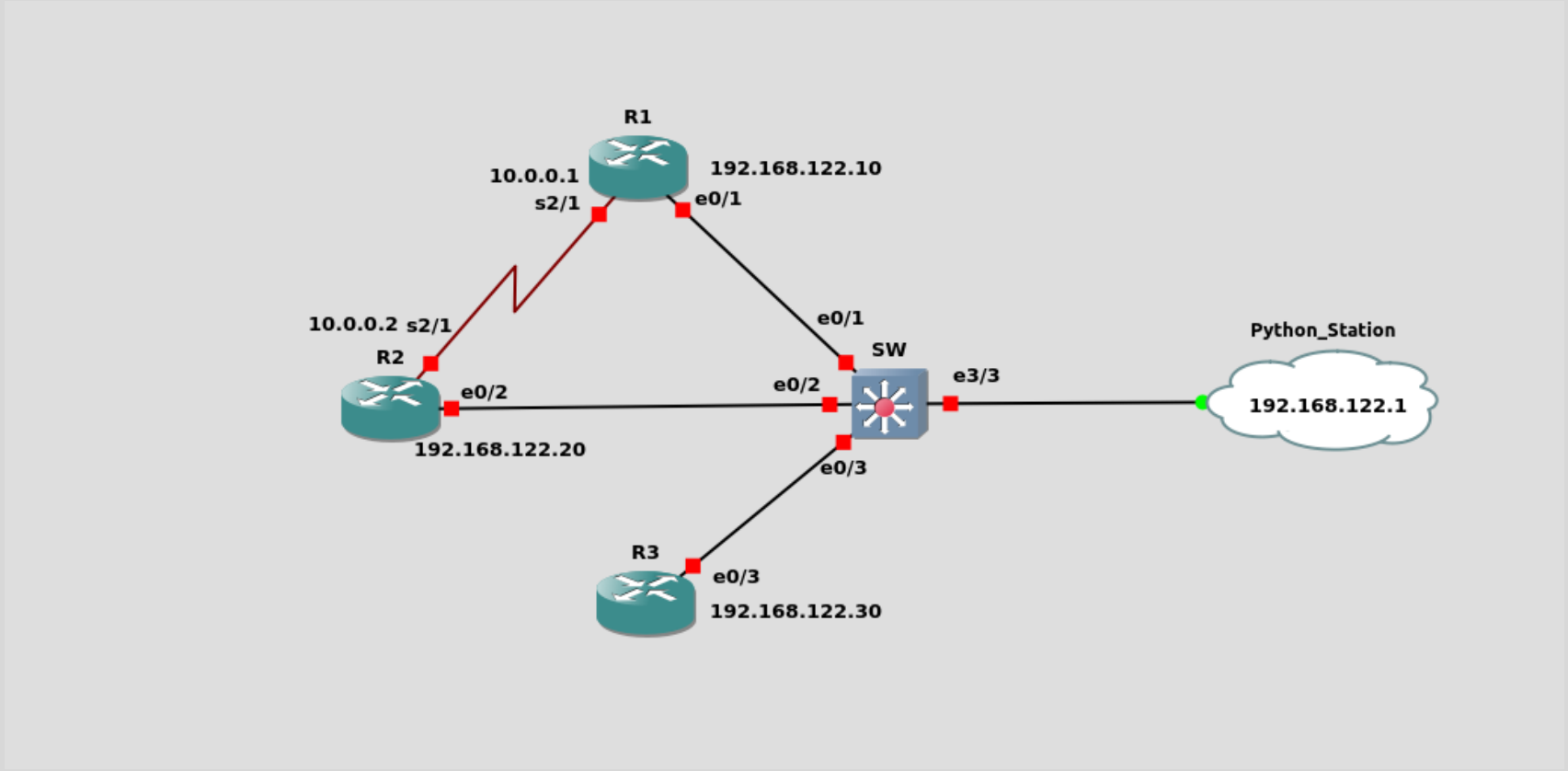
Introduction

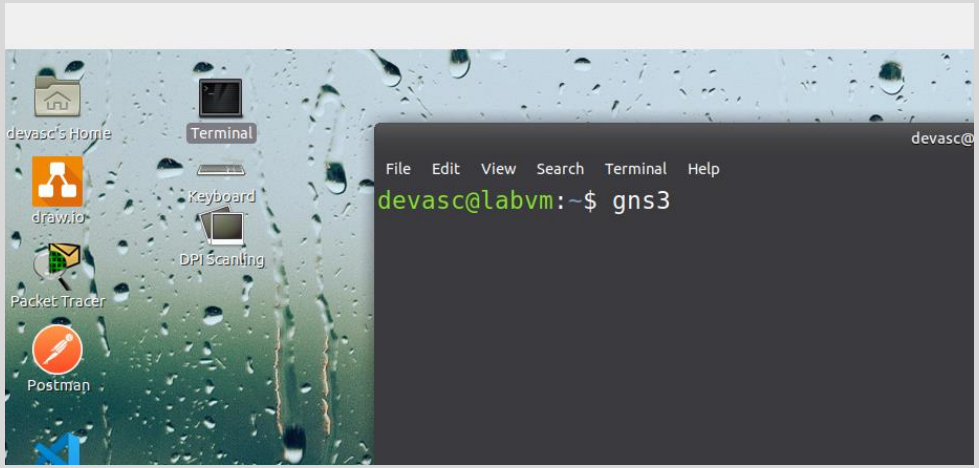
- WE SAW THAT PARAMIKO LIBRARY USED TO AUTOMATE CONFIGURATION TASKS OF NETWORKING DEVICES USING SSH.
- NETMIKO IS A NICER LIBRARY SUPPORT MULTI VENDOR NETWORK LIBRARY BASED ON PARAMIKO.
- IT RUNS ON TOP OF PARAMIKO AND IS USED TO REDUCE ITS COMPLEXITY.
- BOTH PARAMIKO AND NETMIKO ARE ALTERNATIVES TO CONFIGURE DEVICES THAT DO NOT SUPPORT APIS.
- AN API IS A STRUCTURED MODE OF SENDING AND RECEIVING STRUCTURED DATA FROM NETWORK DEVICES.

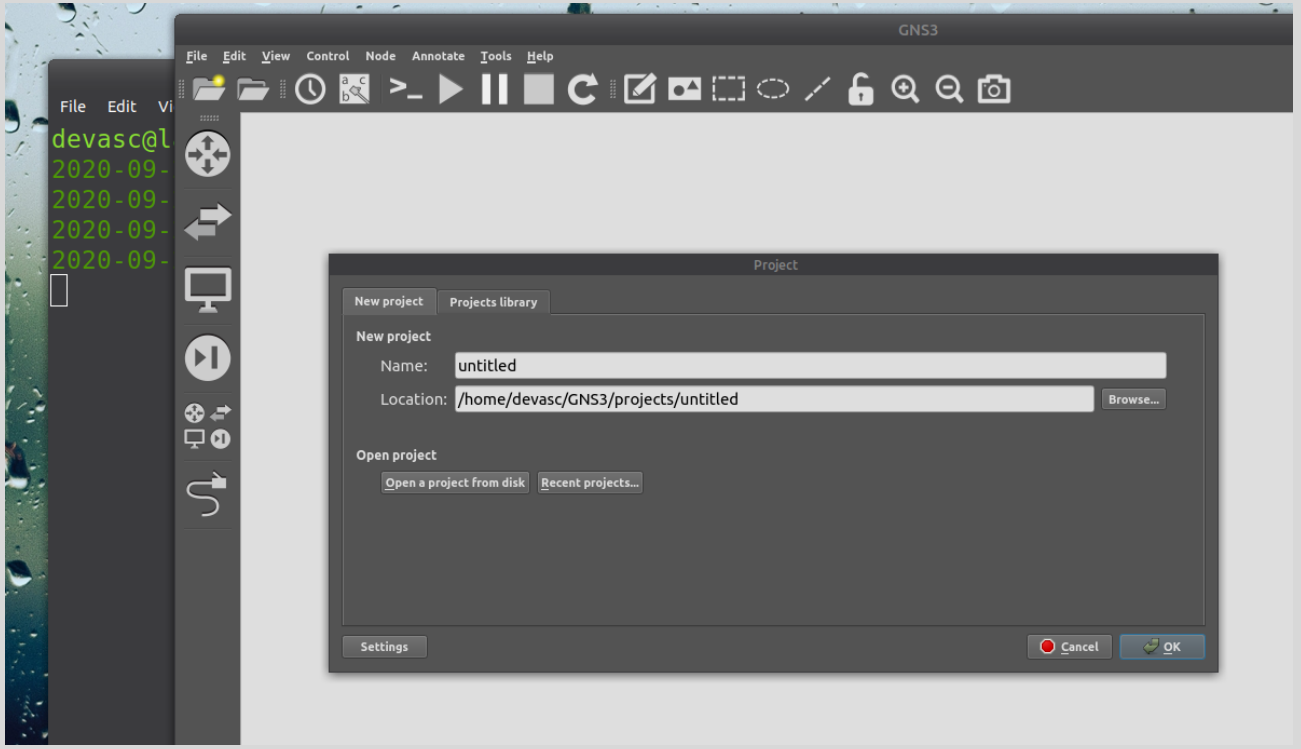
Paramiko vs. Netmiko.

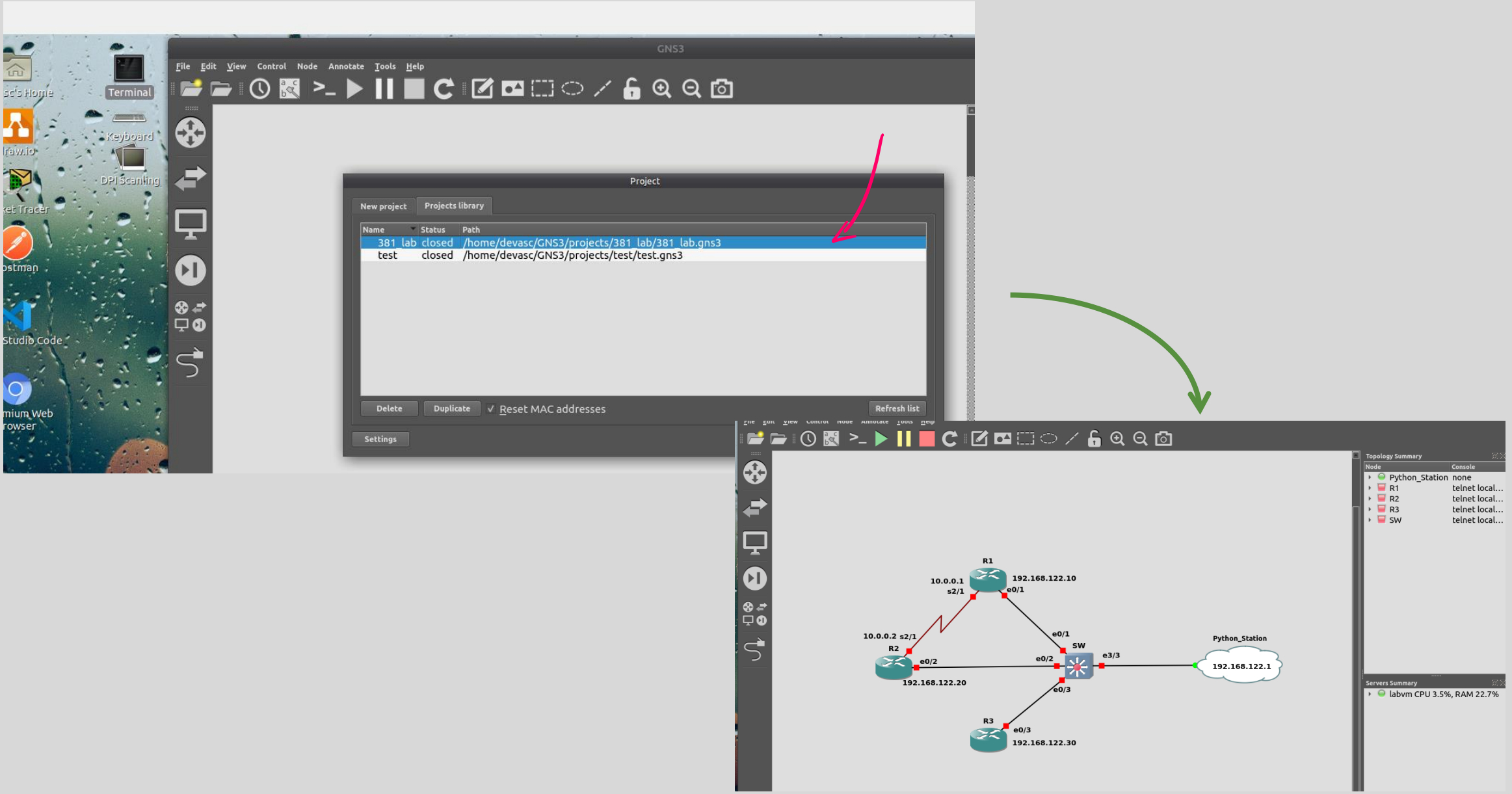
- Paramiko can be used to communicate with any device that supports ssh.
- Although Netmiko is easier to use than Paramiko, it supports only some devices.
- It supports however the most important and used vendors.
- If the platform is supported we could choose Netmiko because it's easier to handle.
- We write less code and reduce the possibility of having errors.
- If a devices that’s not being supported by Netmiko we could go ahead with Paramiko.

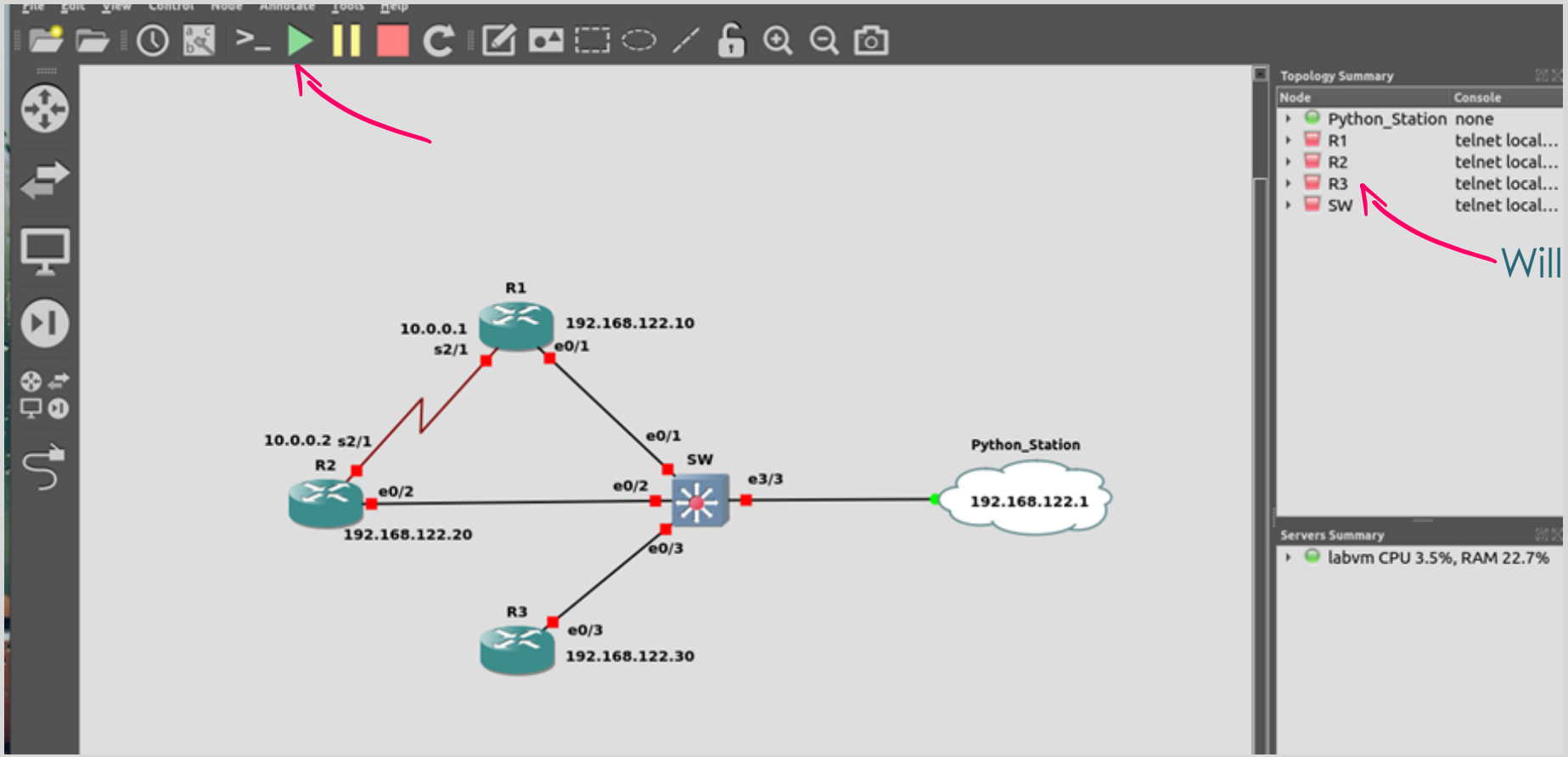
GNS3 Project



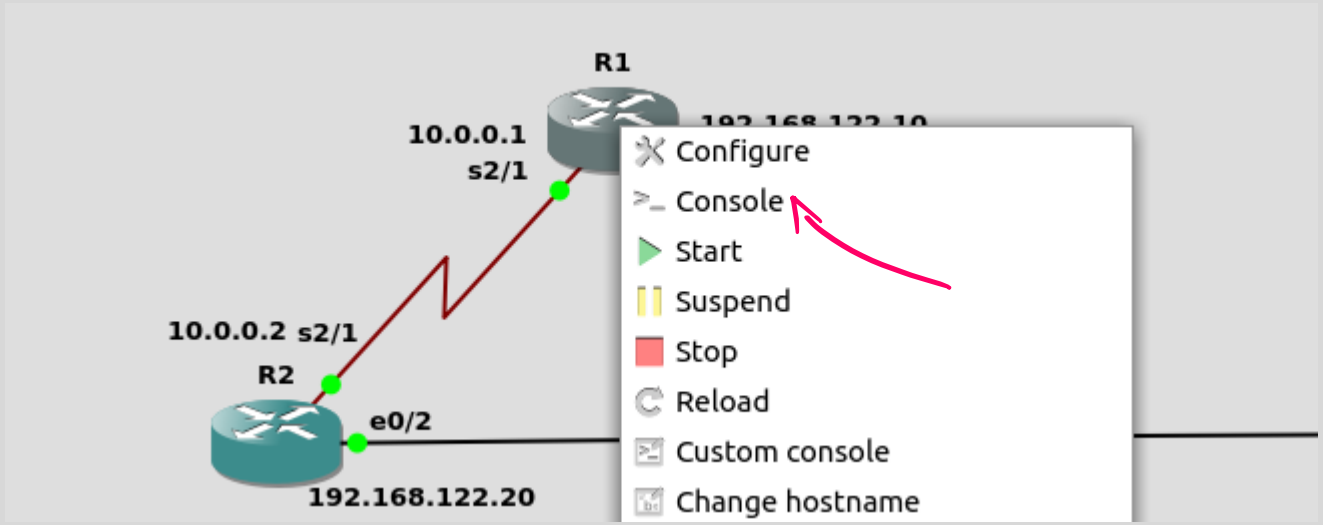

devasc's Home
Terminal
draw.io
Keyboard
DPI Scaling
Packet Tracer
Postman
devasc@
File Edit View Search Terminal Help
devasc@labvm:~$ gns3


GNS3
File Edit View Control Node Annotate Tools Help
File Edit Vi
devasc@l
2020-09-
2020-09-
2020-09-
2020-09-
Project
New project
Projects library
New project
Name: untitled
Location: /home/devasc/GNS3/projects/untitled
Browse...
Open project
Open a project from disk
Recent projects...
Settings
Cancel
OK


GNS3
File Edit View Control Node Annotate Tools Help
Terminal
Keyboard
DPI Scanling
Project
New project
Projects library
Name Status Path
381_lab closed /home/devasc/GNS3/projects/381_lab/381_lab.gns3
test closed /home/devasc/GNS3/projects/test/test.gns3
Delete
Duplicate
Reset MAC addresses
Refresh list
Settings
Topology Summary
Node Console
Python_Station none
R1 telnet local...
R2 telnet local...
R3 telnet local...
SW telnet local...
Servers Summary
labvm CPU 3.5%, RAM 22.7%
R1
10.0.0.1
s2/1
192.168.122.10
e0/1
10.0.0.2 s2/1
R2
e0/2
192.168.122.20
e0/1
SW
e0/2
e3/3
Python_Station
192.168.122.1
e0/3
R3
e0/3
192.168.122.30



Will go Green




```
R1
File  Edit  View  Search  Terminal  Help
*Sep 29 19:57:44.542: %LINK-5-CHANGED: Interface Ethernet
0/2, changed state to administratively down
*Sep 29 19:57:44.630: %LINK-5-CHANGED: Interface Ethernet
0/3, changed state to administratively down
*Sep 29 19:57:44.630: %LINK-5-CHANGED: Interface Ethernet
1/0, changed state to administratively down
*Sep 29 19:57:44.630: %LINK-5-CHANGED: Interface Ethernet
1/1, changed state to administratively down
*Sep 29 19:57:44.630: %LINK-5-CHANGED: Interface Ethernet
1/2, changed state to administratively down
*Sep 29 19:57:44.630: %LINK-5-CHANGED: Interface Ethernet
1/3, changed state to administratively down
*Sep 29 19:57:44.630: %LINK-5-CHANGED: Interface Serial2/
0, changed state to administratively down
*Sep 29 19:57:44.721: %LINK-5-CHANGED: Interface Serial2/
1, changed state to administratively down
IOU1#
```

```
from netmiko import Netmiko

connection = Netmiko(host='192.168.122.10', port='22', username='cisco', password='cisco',
                     device_type='cisco_ios')

output = connection.send_command('sh ip int brief')
print(output)

print('Closing connection')
connection.disconnect()
print('#'*40)
```

Netmiko class

Support dev

output

Close Conn

```
Interface              IP-Address      OK?   Method  Status                  Protocol
Ethernet0/0            unassigned      YES   NVRAM   administratively down   down
Ethernet0/1            192.168.122.10  YES   NVRAM   up                      up
.....
Closing connection
########################################
```

```
from netmiko import Netmiko

connection = Netmiko(host='192.168.122.10', port='22', username='cisco', password='cisco',
                     device_type='cisco_ios')

output = connection.send_command('sh ip int brief')
print(output)

print('Closing connection')
connection.disconnect()
print('#'*40)
```

VS

```
import paramiko
import time
ssh_client = paramiko.SSHClient() # creating an ssh client object
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
router = {'hostname': '192.168.122.10','port': '22', 'username':'cicso', 'password':'cisco'}
ssh_client.connect(**router, look_for_keys=False, allow_agent=False)
print(f'Connecting to {router["hostname"]}')

shell = ssh_client.invoke_shell()
shell.send('show version\n')

time.sleep(1)
output = shell.recv(10000)
print(output)

if print(ssh_client.get_transport().is_active()) == True:
    print('Closing connection')
    ssh_client.close()
```

```
Interface              IP-Address      OK?   Method  Status                  Protocol
Ethernet0/0            unassigned      YES   NVRAM   administratively down   down
Ethernet0/1            192.168.122.10  YES   NVRAM   up                      up
.....
Closing connection
########################################
```

```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                'secret': 'cisco', 'verbose': True}

connection = ConnectHandler(**router)
output = connection.send_command('sh run')
print(output)

print('Closing connection')
connection.disconnect()
print('#'*40)
```

→ sh run instead

```
Interactive SSH session established
^
% Invalid input detected at '^' marker.

Closing connection
########################################
```

enter enable mode

```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                    'secret': 'cisco', 'verbose': True}

connection = ConnectHandler(**router)
connection.enable()
output = connection.send_command('sh run')
print(output)

print('Closing connection')
connection.disconnect()
print('#'*40)
```

```
transport input telnet ssh
!
!
end
Closing connection
########################################
```


```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                'secret': 'cisco', 'verbose': True}

connection = ConnectHandler(**router)
prompt = connection.find_prompt()
if '>' in prompt:
    connection.enable()

output = connection.send_command('sh run')
print(output)

print('Closing connection')
connection.disconnect()
print('#'*40)
```

Show current mode

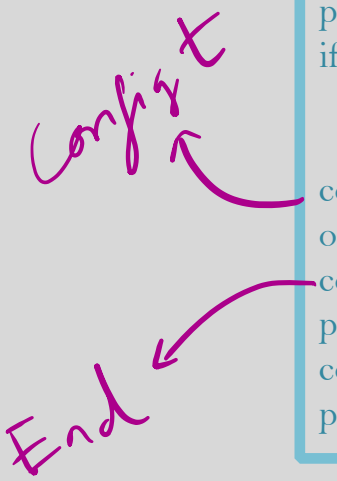
```
transport input telnet ssh
!
!
end
Closing connection
########################################
```

```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                    'secret': 'cisco', 'verbose': True}

connection = ConnectHandler(**router)
prompt = connection.find_prompt()
if '>' in prompt:
    connection.enable()

output = connection.send_command('userna cisco1 secret cisco')
print(output)
print('Closing connection')
connection.disconnect()
print('#'*40)
```

create username

```
Interactive SSH session established
^
% Invalid input detected at '^' marker.

Closing connection
########################################
```

```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                    'secret': 'cisco', 'verbose': True}
connection = ConnectHandler(**router)
prompt = connection.find_prompt()
if '>' in prompt:
    connection.enable()

connection.config_mode()
output = connection.send_command('userna cisco1 secret cisco')
connection.exit_config_mode()
print('Closing connection')
connection.disconnect()
print('#'*40)
```




```
Closing connection
########################################
```

```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                'secret': 'cisco', 'verbose': True}
connection = ConnectHandler(**router)
prompt = connection.find_prompt()
if '>' in prompt:
    connection.enable()

connection.config_mode()
output = connection.send_command('int lo 0')
connection.exit_config_mode()
print('Closing connection')
connection.disconnect()
print('#'*40)
```

→ create Lo 0

nothing → freeze

```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                    'secret': 'cisco', 'verbose': True}
connection = ConnectHandler(**router)
prompt = connection.find_prompt()
if '>' in prompt:
    connection.enable()

connection.config_mode()
output = connection.send_command('int lo 0')
connection.exit_config_mode()
print('Closing connection')
connection.disconnect()
print('#'*40)
```

ctrl-c

```
File "/home/devasc/.local/lib/python3.8/site-packages/netmiko/base_connection.py", line 1425,
in send_command
    time.sleep(delay_factor * loop_delay)
KeyboardInterrupt
```

keep sleeping

```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                'secret': 'cisco', 'verbose': True}
connection = ConnectHandler(**router)
prompt = connection.find_prompt()
if '>' in prompt:
    connection.enable()

connection.config_mode()
output = connection.send_config_set('int lo 0', 'exit')
connection.exit_config_mode()
print('Closing connection')
connection.disconnect()
print('#'*40)
```

get out of int config mode

```
Closing connection
########################################
```

```
from netmiko import ConnectHandler

router = {'device_type': 'cisco_ios', 'host': '10.1.1.10', 'username': 'u1','password': 'cisco','port': 22,
                                'secret': 'cisco', 'verbose': True}
connection = ConnectHandler(**router)
prompt = connection.find_prompt()
if '>' in prompt:
    connection.enable()
if  not connection.check_config_mode()
    connection.config_mode()
output = connection.send_config_set('int lo 0', 'exit')
connection.exit_config_mode()
print('Closing connection')
connection.disconnect()
print('#'*40)
```

→ not in global config !!!

```
Closing connection
########################################
```

Command Sets

```
……..
cmd = '''int lo 0
ip add 1.1.1.1 255.255.255.255
exit
username cisco1 secret cisco
'''
if  not connection.check_config_mode():
        connection.config_mode()

connection.send_config_set(cmd.split('\n'))
connection.exit_config_mode()

print('Closing connection')
connection.disconnect()
print('#'*40)
```

Split by newline

```
Closing connection
########################################
```

```
router ospf  1
router-id 1.1.1.1
net 0.0.0.0 0.0.0.0 area 0
distance 80
default-information originate
```

Save to 192.168.122.10_ospf.txt

```
……..
connection = ConnectHandler(**router)
prompt = connection.find_prompt()
if '>' in prompt:
    connection.enable()

print('Sending commands from file ...')
output = connection.send_config_from_file('192.168.122.10_ospf.txt')
print(output)

print('Closing connection')
connection.disconnect()
print('#'*40)
```

Read commands from file

```
R1#
Closing connection
########################################
```

```
router ospf 1
router-id 1.1.1.1
net 0.0.0.0 0.0.0.0 area 0
distance 80
default-information originate
```

Save to 192.168.122.10_ospf.txt

```
router ospf 1
router-id 2.2.2.2
net 0.0.0.0 0.0.0.0 area 0
distance 80
default-information originate
```

192.168.122.20_ospf.txt

```
router ospf 1
router-id 3.3.3.3
net 0.0.0.0 0.0.0.0 area 0
distance 80
default-information originate
```

192.168.122.30_ospf.txt

```
[{
    'device_type': 'cisco_ios',
    'host': '192.168.122.10',
    'username': 'cisco',
    'password': 'cisco',
    'port': 22,
    'secret': 'cisco',
    'verbose': True
},
{
    'device_type': 'cisco_ios',
    'host': '192.168.122.20',
    'username': 'cisco',
    'password': 'cisco',
    'port': 22,
    'secret': 'cisco',
    'verbose': True
},
{
    'device_type': 'cisco_ios',
    'host': '192.168.122.30',
    'username': 'cisco',
    'password': 'cisco',
    'port': 22,
    'secret': 'cisco',
    'verbose': True
}]
```

Save to routers.txt

```
from netmiko import ConnectHandler
import myNewParamiko as m

routers = m.get_list_from_file ('routers.txt')
for router in routers:
    connection = ConnectHandler(**router)
    prompt = connection.find_prompt()
    if '>' in prompt:
        connection.enable()
    print('Sending commands from file ...')
    output = connection.send_config_from_file(router['host']+'_ospf.txt')
    print(output)
    print('Closing connection')
    connection.disconnect()
    print('#'*40)
```

192.168.122.xx_ospf.txt

```
R3#
Closing connection
########################################
```

```
from netmiko import ConnectHandler
import myNewParamiko as m
import time
start = time.time()
routers = m.get_list_from_file ('routers.txt')
for router in routers:
    connection = ConnectHandler(**router)
    prompt = connection.find_prompt()
    if '>' in prompt:
        connection.enable()
    print('Sending commands from file ...')
    output = connection.send_config_from_file(router['host']+'_ospf.txt')
    print(output)
    print('Closing connection')
    connection.disconnect()
    print('#'*40)
end = time.time()
print(f'Total execution time:{end-start}')
```

time before run

time after run

accumulation time

```
R3#
Closing connection
########################################
Total execution time:28.53866672515869
```

~ 9 x 3 seconds

```
from netmiko import ConnectHandler
import myNewParamiko as m
import time
import threading

start = time.time()
def config_ospf(router):
    connection = ConnectHandler(**router)
    prompt = connection.find_prompt()
    if '>' in prompt:
        connection.enable()
    print('Sending commands from file ...')
    output = connection.send_config_from_file(router['host']+'_ospf.txt')
    print(output)
    print('Closing connection')
    connection.disconnect()
    print('#'*40)

routers = m.get_list_from_file ('routers.txt')
threads = list()
for router in routers:
    th = threading.Thread(target=config_ospf, args=(router,))
    threads.append(th)

for th in threads:
    th.start()

for th in threads:
    th.join()

end = time.time()
print(f'Total execution time:{end-start}')
```

make a function

task( )

Split to 3 threads

```
R2(config-router)#end
R2#
Closing connection
##########################################
##########################################
##########################################
Total execution time:9.560632228851318
```

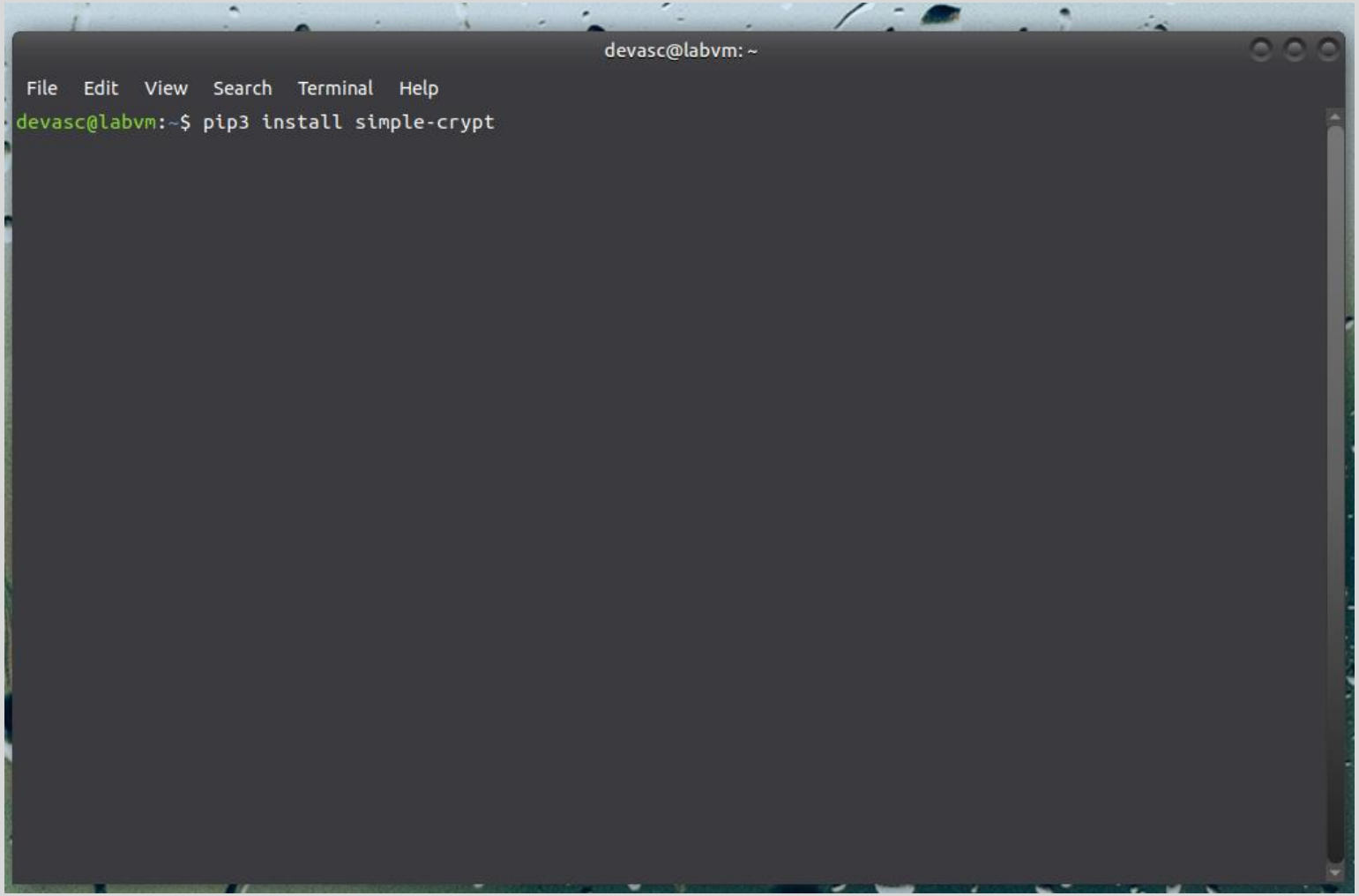
√s 29s

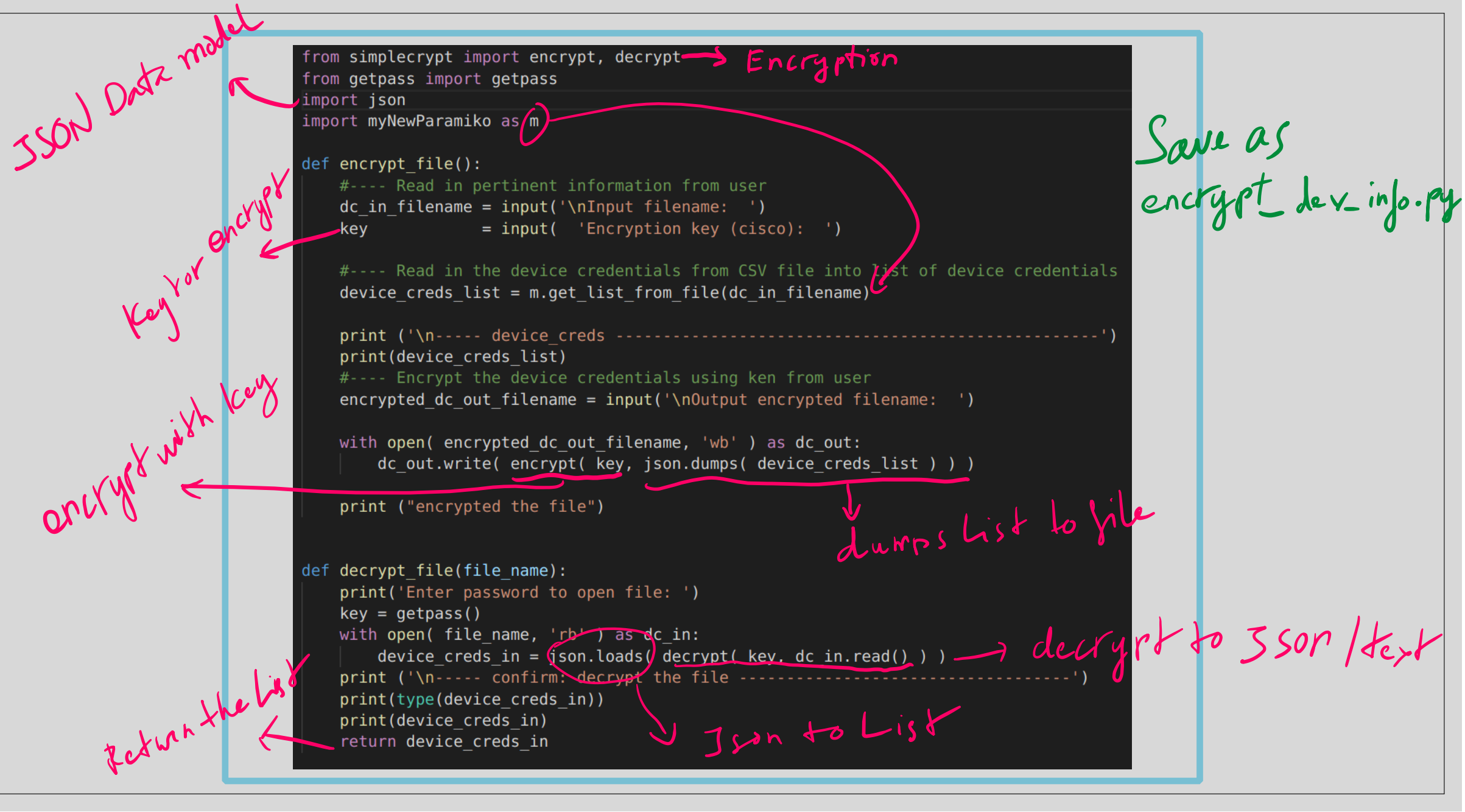
```
[{
    'device_type': 'cisco_ios',
    'host': '192.168.122.10',
    'username': 'cisco',
    'password': 'cisco',
    'port': 22,
    'secret': 'cisco',
    'verbose': True
},
{
    'device_type': 'cisco_ios',
    'host': '192.168.122.20',
    'username': 'cisco',
    'password': 'cisco',
    'port': 22,
    'secret': 'cisco',
    'verbose': True
},
{
    'device_type': 'cisco_ios',
    'host': '192.168.122.30',
    'username': 'cisco',
    'password': 'cisco',
    'port': 22,
    'secret': 'cisco',
    'verbose': True
}]
```

clear text

We need encryption

Open terminal and install simple-crypt






```python
from simplecrypt import encrypt, decrypt
from getpass import getpass
import json
import myNewParamiko as m

def encrypt_file():
    #---- Read in pertinent information from user
    dc_in_filename = input('\nInput filename:  ')
    key            = input(  'Encryption key (cisco):  ')

    #---- Read in the device credentials from CSV file into list of device credentials
    device_creds_list = m.get_list_from_file(dc_in_filename)

    print ('\n----- device_creds ------------------------------------------------------')
    print(device_creds_list)
    #---- Encrypt the device credentials using ken from user
    encrypted_dc_out_filename = input('\nOutput encrypted filename:  ')

    with open( encrypted_dc_out_filename, 'wb' ) as dc_out:
        dc_out.write( encrypt( key, json.dumps( device_creds_list ) ) )

    print ("encrypted the file")


def decrypt_file(file_name):
    print('Enter password to open file: ')
    key = getpass()
    with open( file_name, 'rb' ) as dc_in:
        device_creds_in = json.loads( decrypt( key, dc_in.read() ) )
    print ('\n----- confirm: decrypt the file -----------------------------------')
    print(type(device_creds_in))
    print(device_creds_in)
    return device_creds_in
```

Open Terminal

```
devasc@labvm:~/labs/devnet-src/sample-app$ python3
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import encrypt_dev_info as edi
>>> edi.encrypt_file()

Input filename:  routers.txt
Encryption key (cisco):  cisco

----- device_creds -----------------------------------------------------
[{'device_type': 'cisco_ios', 'host': '192.168.122.10', 'username': 'cisco', 'password': 'cisco', 'port': 22, 'secret':
 'cisco', 'verbose': True}, {'device_type': 'cisco_ios', 'host': '192.168.122.20', 'username': 'cisco', 'password': 'ci
sco', 'port': 22, 'secret': 'cisco', 'verbose': True}, {'device_type': 'cisco_ios', 'host': '192.168.122.30', 'username
': 'cisco', 'password': 'cisco', 'port': 22, 'secret': 'cisco', 'verbose': True}]

Output encrypted filename:  encrypted_routers.txt
encrypted the file
>>>
devasc@labvm:~/labs/devnet-src/sample-app$ cat encrypted_routers.txt
sc[C#P◆<◆◆N~◆t◆t]+)◆5◆!◆w◆K◆R◆#◆
```

→ file needs to encrypt

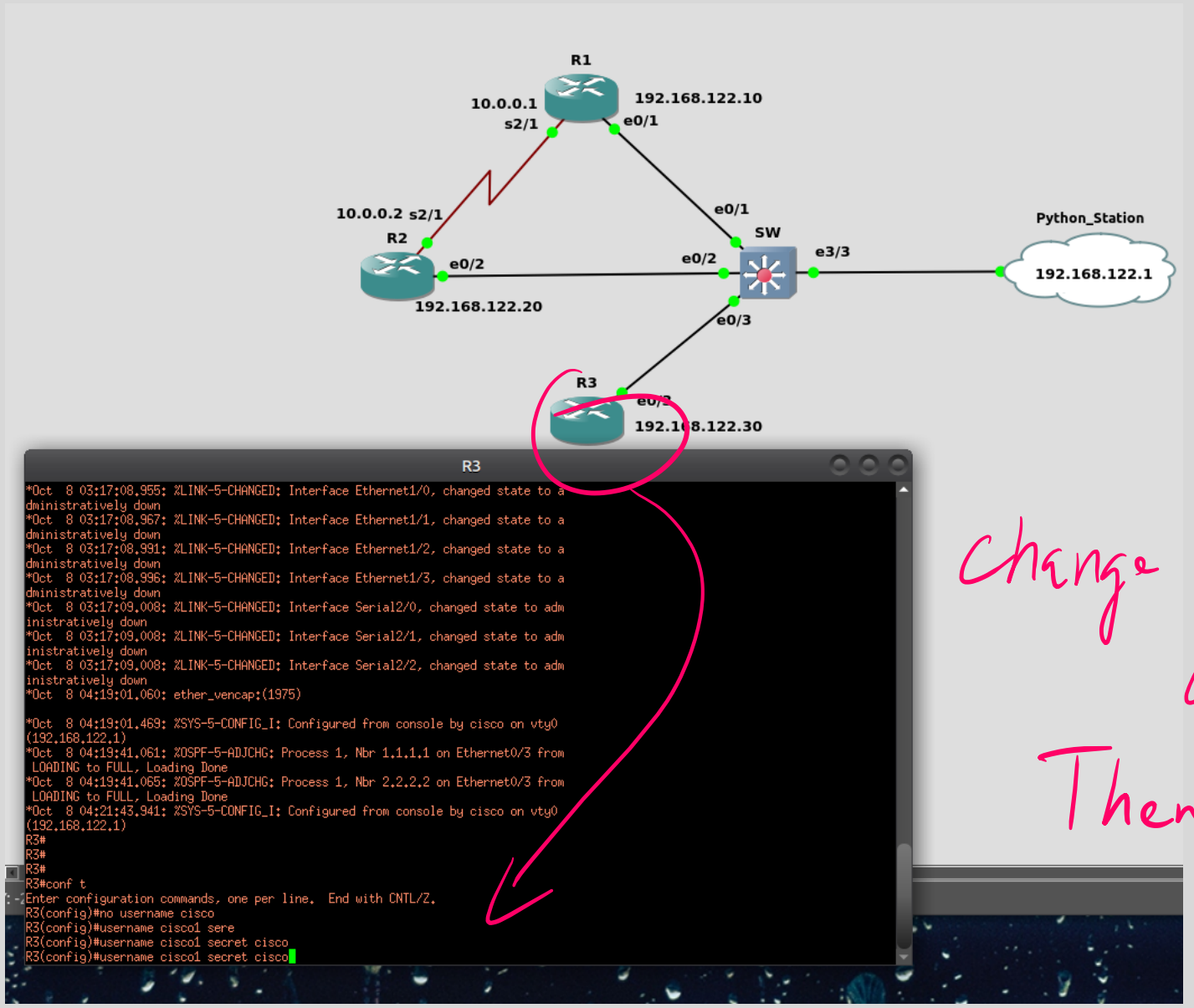
→ key

→ read the encrypted file

```python
from netmiko import ConnectHandler
import encrypt_dev_info as edi
import time
import threading

start = time.time()
def config_ospf(router):
    connection = ConnectHandler(**router)
    prompt = connection.find_prompt()
    if '>' in prompt:
        connection.enable()
    print('Sending commands from file ...')
    output = connection.send_config_from_file(router['host']+'_ospf.txt')
    print(output)
    print('Closing connection')
    connection.disconnect()
    print('#'*40)

encrypte_file = input('Enter encrypted file: ')
routers = edi.decrypt_file (encrypte_file)
threads = list()
for router in routers:
    th = threading.Thread(target=config_ospf, args=(router,))
    threads.append(th)

for th in threads:
    th.start()

for th in threads:
    th.join()

end = time.time()
print(f'Total execution time:{end-start}')
```

key

```
Enter encrypted file: encrypted_routers.txt
Enter password to open file:
Password:

----- confirm: decrypt the file --------------------------------
<class 'list'>
[{'device_type': 'cisco_ios', 'host': '192.168.122.10', 'username': 'cisco', 'password': 'cisco', 'port': 22,
'secret': 'cisco', 'verbose': True}, {'device_type': 'cisco_ios', 'host': '192.168.122.20', 'username': 'cisco',
'password': 'cisco', 'port': 22, 'secret': 'cisco', 'verbose': True}, {'device_type': 'cisco_ios', 'host':
'192.168.122.30', 'username': 'cisco', 'password': 'cisco', 'port': 22, 'secret': 'cisco', 'verbose': True}]
SSH connection established to 192.168.122.20:22
Interactive SSH session established
SSH connection established to 192.168.122.10:22
Interactive SSH session established
SSH connection established to 192.168.122.30:22
Interactive SSH session established
Sending commands from file ...
Sending commands from file ...
Sending commands from file ....……..
……..
……..
Closing connection
##########################################
##########################################
##########################################
Total execution time:9.560632228851318
```



Change username to cisco1

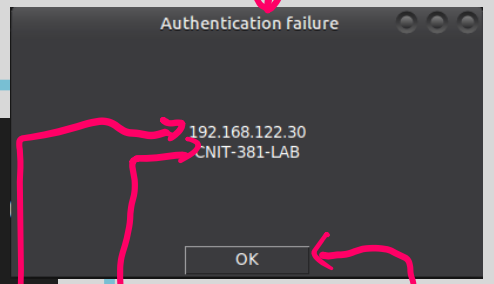
Then run again

```
Traceback (most recent call last):
  File "/home/devasc/.local/lib/python3.8/site-packages/netmiko/base_connection.py", line 899, in establish_connection
    self.remote_conn_pre.connect(**ssh_connect_params)
  File "/home/devasc/.local/lib/python3.8/site-packages/paramiko/client.py", line 435, in connect
    self._auth(
  File "/home/devasc/.local/lib/python3.8/site-packages/paramiko/client.py", line 764, in _auth
    raise saved_exception
  File "/home/devasc/.local/lib/python3.8/site-packages/paramiko/client.py", line 751, in _auth
    self._transport.auth_password(username, password)
  File "/home/devasc/.local/lib/python3.8/site-packages/paramiko/transport.py", line 1509, in auth_password
    return self.auth_handler.wait_for_response(my_event)
  File "/home/devasc/.local/lib/python3.8/site-packages/paramiko/auth_handler.py", line 250, in wait_for_response
    raise e
paramiko.ssh_exception.AuthenticationException: Authentication failed.
```

App stops → How to skip errors and keep running ???

alert_popup.py

```python
import tkinter as tk

def alert_popup(title, message, path):
    """Generate a pop-up window for special messages."""
    root = tk.Tk()
    root.title(title)
    w = 400     # popup window width
    h = 200     # popup window height
    sw = root.winfo_screenwidth()
    sh = root.winfo_screenheight()
    x = (sw - w)/2
    y = (sh - h)/2
    root.geometry('%dx%d+%d+%d' % (w, h, x, y))
    m = message
    m += '\n'
    m += path
    w = tk.Label(root, text=m, width=120, height=10)
    w.pack()
    b = tk.Button(root, text="OK", command=root.destroy, width=10)
    b.pack()
    tk.mainloop()
```



Catch exceptions and keep running

Exceptions

alert popup

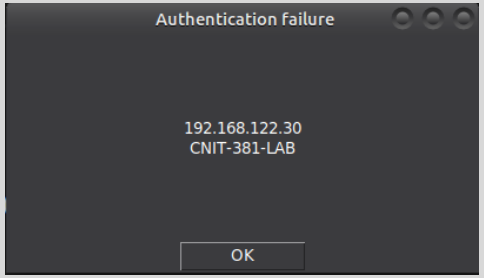
New!

popups

```python
from netmiko import ConnectHandler
from netmiko.ssh_exception import NetMikoTimeoutException
from paramiko.ssh_exception import SSHException
from netmiko.ssh_exception import AuthenticationException
import encrypt_dev_info as ed
import alert_popup as ap
import time
import threading

start = time.time()
def config_ospf(router):
    try:
        connection = ConnectHandler(**router)
    except (AuthenticationException):
        ap.alert_popup('Authentication failure', router['host'],'CNIT-381-LAB')
        return
    except (NetMikoTimeoutException):
        ap.alert_popup('Timeout to device', router['host'],'CNIT-381-LAB')
        return
    except (SSHException):
        ap.alert_popup('SSH Issue. Are you sure SSH is enabled?', router['host'],'CNIT-381-LAB')
        return
    except Exception as unknown_error:
        ap.alert_popup('Some other error', str(unknown_error), '')
        return

    prompt = connection.find_prompt()
    if '>' in prompt:
        connection.enable()
    print('Sending commands from file ...')
    output = connection.send_config_from_file(router['host']+'_ospf.txt')
    print(output)
    print('Closing connection')
    connection.disconnect()
    print('#'*40)
```

let's run it

. . . . . .

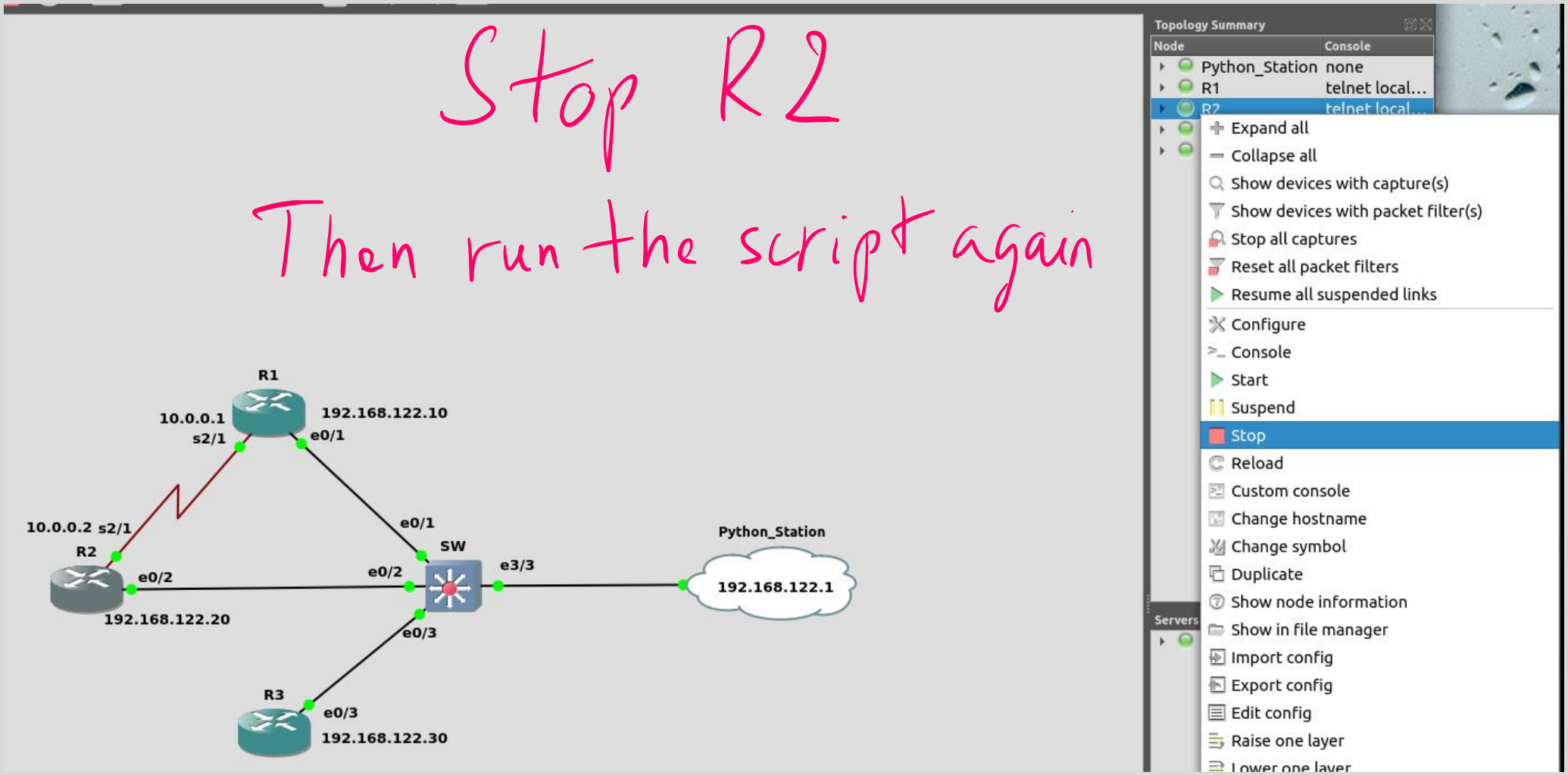



```
R2(config-router)#default-information originate
R2(config-router)#end
R2#
Closing connection
config term
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf  1
R1(config-router)#router-id 1.1.1.1
R1(config-router)#net 0.0.0.0 0.0.0.0 area 0
R1(config-router)#distance 80
R1(config-router)#default-information originate
R1(config-router)#end
R1#
Closing connection
##########################################
##########################################
Total execution time:36.40194511413574
```

No error

Stop R2

Then run the script again



many threads (message boxes) are accessing to 1 thread the UI thread or Main thread)

. . . . . .

```
 raise NetmikoTimeoutException(msg)
netmiko.ssh_exception.NetmikoTimeoutException: Connection to device timed-out: cisco_ios
192.168.122.10:22

During handling of  the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/lib/python3.8/threading.py", line 932, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3.8/threading.py", line 870, in run
    self._target(*self._args, **self._kwargs)
  File "/home/devasc/labs/devnet-src/sample-app/netmiko_8.py", line 19, in config_ospf
    ap.alert_popup('Timeout to device', router['host'],'CNIT-381-LAB')
  File "/home/devasc/labs/devnet-src/sample-app/alert_popup.py", line 22, in alert_popup
    tk.mainloop()
  File "/usr/lib/python3.8/tkinter/__init__.py", line 593, in mainloop
    _default_root.tk.mainloop(n)
RuntimeError: Calling Tcl from different apartment
Total execution time:72.54487824440002
```

```python
from netmiko import ConnectHandler
from netmiko.ssh_exception import NetMikoTimeoutException
from paramiko.ssh_exception import SSHException
from paramiko.ssh_exception import NoValidConnectionsError
from netmiko.ssh_exception import AuthenticationException
import encrypt_dev_info as ed
import time
import threading

start = time.time()
def config_ospf(router):
    try:
        connection = ConnectHandler(**router)
    except (AuthenticationException):
        print('Authentication failure'+ router['host']+ 'CNIT-381-LAB')
        return
    except (NetMikoTimeoutException):
        print('Timeout'+ router['host']+ 'CNIT-381-LAB')
        return
    except (SSHException):
        print('Authentication failure'+ router['host']+ 'CNIT-381-LAB')
        return
    except (NoValidConnectionsError):
        print('Novalid'+ router['host']+ 'CNIT-381-LAB')
        return
    except Exception as unknown_error:
        print('Unknown failure'+ router['host']+ 'CNIT-381-LAB')
        return

    prompt = connection.find_prompt()
    if '>' in prompt:
        connection.enable()
    print('Sending commands from file ...')
    output = connection.send_config_from_file(router['host']+'_ospf.txt')
    print(output)
    print('Closing connection')
    connection.disconnect()
    print('#'*40)

encrypte_file = input('Enter encrypted file: ')
routers = ed.decrypt_file (encrypte_file)
threads = list()
for router in routers:
    th = threading.Thread(target=config_ospf, args=(router,))
    threads.append(th)

for th in threads:
    th.start()

for th in threads:
    th.join()

end = time.time()
print(f'Total execution time:{end-start}')
```

Remove popups

Print instead

```
SH connection established to 192.168.122.20:22
Interactive SSH session established
Timeout 192.168.122.20 CNIT-381-LAB
Authentication failure 192.168.122.30 CNIT-381-LAB
Sending commands from file ...
config term
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router ospf  1
R2(config-router)#router-id 2.2.2.2
R2(config-router)#net 0.0.0.0 0.0.0.0 area 0
R2(config-router)#distance 80
R2(config-router)#default-information originate
R2(config-router)#end
R2#
Closing connection
##########################################
Total execution time:27.386638641357422
```