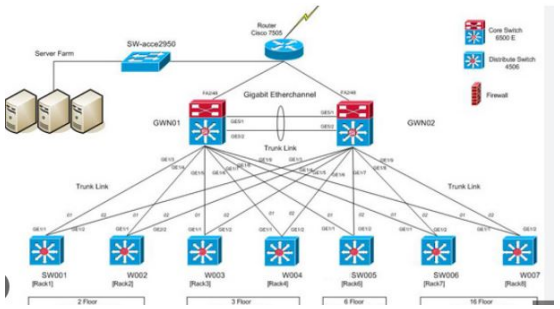
PROGRAMMIEREN NETZWERKE MIT PYTHON MODULE,NETMIKO,NAPALAM

**WRITTEN BEI :DUNYA ABDULRAZZAQ

PYTHON
FOR NETWORK ENGINEERS

INHALT



PROJEKTÜBERSICHT:

Netzwerkingenieure oder Administrator können Automatisierung nutzen, um sich weniger auf manuelle Eingaben zu verlassen und sicherzustellen, dass ihre Netzwerke mit weniger Fehlern funktionieren. Python ist eine der am meisten bevorzugten Sprachen für die Implementierung von Programmierung und Automatisierung . Netzwerkteams können damit einfache Skriptes schreiben, die Aufgaben wie Netzwerkerkennung, Gerätekonfiguration und Fehlerbehebung automatisieren.

Das Projekt zielt darauf ab, die Nutzung dieser Module ohne die Notwendigkeit einer weiteren Konfigurationsmanagement-Plattform wie Ansible, Chef oder Puppet zu ermöglichen. Stattdessen soll ein einfaches Programm mit einer grafischen Benutzeroberfläche (GUI) auf Basis von Tkinter verwendet werden, um Verwaltungs- und Konfigurationsaufgaben zu erleichtern. Dieses Programm ist besonders für kleine und mittelgroße Netzwerke gedacht, in denen möglicherweise ein Managementsystem fehlt.

EINFÜHRUNG ZU AUTOMATISIERUNG IN PYTHON

Tools wie NAPALM und Netmiko erleichtern die Konfiguration und Interaktion mit Netzwerkgeräten mithilfe einer API wie NETCONF oder mithilfe von SSH . Daneben gibt es auch viele andere Python Module,die im Automation Bereich ,ein sehr gute und einfach Lösung anbieten, die Unterschied kommt manchmal bei supported Plattform und beinhaltete Methoden ,auch über welche Protokoll sprechen Sie mit den Infrastruktur .

Neben Netmiko & Napalm gebt es noch mehrere Libraries und Protokoll mit verschieden Einsatz Gebiete bekannte Beispiel,“die hier nur kurz über diesen GNS3 Topology Ihre Ansatz erklärt wurde“ , Telnetlib (use Telnet,) ,Paramiko (SSHv2) ,dazu gibt es noch Restconf, Nornir, Netconf ,Genie ,Nxapi...etc. ,die noch weiter Vertiefung in den Gebiet brauchen können ,und würde hier in diesem Docku oder Programm nicht behandelt .

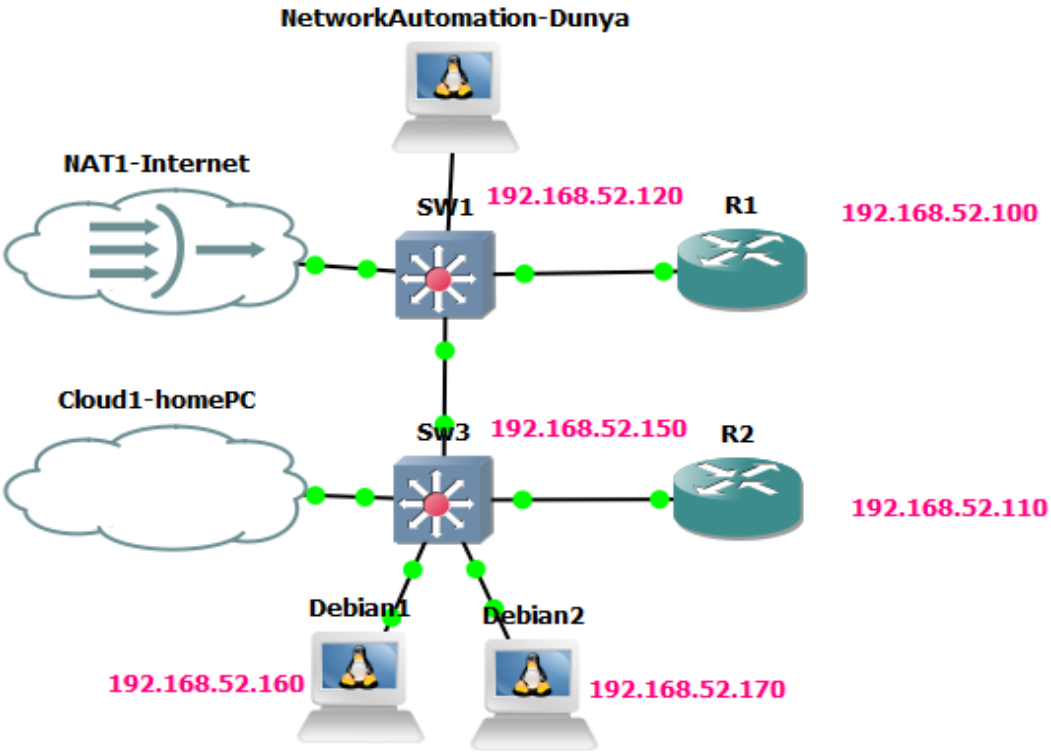
Diese Dokumentation ist nur ein kurz Review für Teile diesen Modulen, mit Screenshots

es noch mehrere Libraries und Protokoll mit verschieden Einsatz Gebiete

INFO&LINK :
https://www.computerweekly.com/de/ratgeber/Netzwerkautomatisierung-mit-Paramiko-Netmiko-und-NAPALM
HTTPS://WWW.CBTNUGGETS.COM/BLOG/TECHNOLOGY/NETWORKING/5-TOP-FREE-PYTHON-LIBRARIES-FOR-NETWORK-AUTOMATION
HTTPS://WWW.CLARIONTECH.COM/BLOG/BEST-PYTHON-MODULES-FOR-AUTOMATION

**In den unten präsentierten Beispielen (im Programm Netzwerk Konfiguration Manager) wurde eine einfache Netzwerkkonfiguration verwendet, um das Konzept ohne Komplexität darzustellen. Es handelt sich hierbei um Automatisierungskonzepte und nicht um fortgeschrittenes Netzwerkdesign.

GNS3 TOPLOGY



TELNETLIB (USING TELNET)

Das Telnetlib-Modul stellt eine Telnet-Klasse bereit, die das Telnet-Protokoll implementiert. Bei Telnetlib weist jedoch einige Unterschiede auf. Der bemerkenswerteste Unterschied besteht darin, dass Telnetlib die Übergabe einer Byte-Zeichenfolge anstelle einer normalen erfordert, dewesgen brauchen wir bei Öffnung der Config Datei der `b` option neben Read&Writt Option,um der Binary Format zu erzeugen .
Mit der Telnetlib von Python können Sie den Zugriff auf Telnet-Server auch von Nicht-Unix-Rechnern problemlos automatisieren. Als flexible Alternative ,Aber bleibt die Sicherheit Risiko Gros ,wegen Unsecure Übertragung ,so es ist nicht empfehlenswert bei Publick oder unsecur Netzwerk Umgebung ,außerdem Telnet ist ein altes Technik .
* die Haupt benutzte Methode bei TelNtLIB : -read_until ,-read ,-write ,-expect ,-close

Beispiele mit Hilfe von oben gestellt Topologie ,Programmierung using Telnetlib :

EX1 :Erstellung von VLANs Hier in diesem kleinen Skript gibt der Benutzer die gewünschte VLAN-Nummer und den dazugehörigen Namen ein, die mithilfe einer Schleife erstellt werden. Die Telnet-Benutzeranmeldedaten werden aus einer JSON-Datei gelesen. Das Programm überprüft bei jedem Gerät die Anmeldeinformationen. Falls diese nicht gespeichert sind (z. B. aus Sicherheitsgründen), wird der Benutzer aufgefordert, sie einzugeben. Passwörter werden dabei über die getpass-Funktion sicher (im verdeckten Modus) eingegeben.

```python
##Switch SW1
import getpass
import telnetlib
import json
import time
with open('device.json', 'r') as f:  #hier wird den json datei,wo steht die Device IP,User,Password, geofnet und über
      user_data = json.load(f)   # if statemnt bei jede item geprüft ,
for item in user_data:           #Falls den User&Password&Enable Passwo nicht im json File eingetragen ,prompet der User
    host = item['Host']          #diese Information zu geben
    user = item['User']          # bei Password wird über getpass in secure art gegeben
    if user != '':
        pass
    else:
        user=input(f'enter User name for {host}:\n>>')
    password = item['Password']
    if password != '':
        pass
    else:
        password = getpass.getpass(prompt=f'bitte enter user {user} Password :\n >>')
    enable_pw = item['Enable']
    if enable_pw != '':
        pass
    else:
        enable_pw = getpass.getpass(prompt=f'bitte enter Enable Password for Host {host} :\n >>')
    tn = telnetlib.Telnet(host)   # Telnet wird hier gebaut

    tn.write(user.encode('ascii') + b'\n')  #username wird gegben, von txt to binary  encoded
    if password:
        tn.read_until(b'Password: ') # warten bis prompet password scheint
        tn.write(password.encode('ascii') + b'\n') #wirde password gegben
    tn.write(b"enable\n")
    tn.write(enable_pw.encode('ascii') + b'\n')
    tn.write(b"conf t\n")
    vlan_no=int(input('bitte geben Sie ,wie viel Vlan wollen Sie erstellen \n'))# wie viel Vlans gewüncht
    for vlan in range(2,vlan_no+1):    #hier wird ein loop zu erstellen Vlans
        tn.write(b'vlan ' + str(vlan).encode('ascii') + b"\n")
        tn.write(b'name VLAN_'+   str(vlan).encode('ascii') + b'\n') #geben ein name mit Vlan no.
    tn.write(b"do wr \n")
    time.sleep(3)
    tn.write(b'end\n')
    tn.write(b'exit\n')
    print(tn.read_all().decode('ascii'))
```

Ex1 :Add VLAN range to Switch sw2(später sw3 gennat) von Automation Docker client , zweites Bild zeigt den Debug im Switch

```
root@NetworkAutomation-Dunya:~/python_project# nano telnet_sw.py
root@NetworkAutomation-Dunya:~/python_project# python3  telnet_sw
bitte enter user name for Host 192.168.52.130 :
 >>cisco
bitte enter user cisco Password :
 >>
bitte enter Enable Password :
 >>
bitte geben Sie ,wie viel Vlan wollen Sie erstellen
7

Sw2>enable
Password:
Sw2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Sw2(config)#vlan 2
Sw2(config-vlan)#name VLAN_2
Sw2(config-vlan)#vlan 3
Sw2(config-vlan)#name VLAN_3
Sw2(config-vlan)#vlan 4
Sw2(config-vlan)#name VLAN_4
Sw2(config-vlan)#vlan 5
Sw2(config-vlan)#name VLAN_5
Sw2(config-vlan)#vlan 6
Sw2(config-vlan)#name VLAN_6
Sw2(config-vlan)#vlan 7
Sw2(config-vlan)#name VLAN_7
Sw2(config-vlan)#end
Sw2#exit

root@NetworkAutomation-Dunya:~/python_project#
```

```
Sw1(config)#do debug telnet
Incoming Telnet debugging is on
Sw1(config)#
*Jun  4 18:07:01.588: Telnet2: 1 1 251 1
*Jun  4 18:07:01.588: TCP2: Telnet sent WILL ECHO (1)
*Jun  4 18:07:01.588: Telnet2: 2 2 251 3
*Jun  4 18:07:01.588: TCP2: Telnet sent WILL SUPPRESS-GA (3)
*Jun  4 18:07:01.588: Telnet2: 80000 80000 253 24
*Jun  4 18:07:01.588: TCP2: Telnet sent DO TTY-TYPE (24)
*Jun  4 18:07:01.588: Telnet2: 10000000 10000000 253 31
*Jun  4 18:07:01.588: TCP2: Telnet sent DO WINDOW-SIZE (31)
*Jun  4 18:07:01.589: TCP2: Telnet received DONT ECHO (1)
*Jun  4 18:07:01.589: TCP2: Telnet sent WONT ECHO (1)
*Jun  4 18:07:01.589: TCP2: Telnet received DONT SUPPRESS-GA (3)
*Jun  4 18:07:01.589: TCP2: Telnet sent WONT SUPPRESS-GA (3)
Sw1(config)#
*Jun  4 18:07:01.589: TCP2: Telnet received WONT TTY-TYPE (24)
*Jun  4 18:07:01.589: TCP2: Telnet sent DONT TTY-TYPE (24)
*Jun  4 18:07:01.589: TCP2: Telnet received WONT WINDOW-SIZE (31)
*Jun  4 18:07:01.589: TCP2: Telnet sent DONT WINDOW-SIZE (31)
*Jun  4 18:07:01.620: TCP2: Telnet received DONT ECHO (1)
*Jun  4 18:07:01.620: TCP2: Telnet received DONT SUPPRESS-GA (3)
*Jun  4 18:07:01.620: TCP2: Telnet received WONT TTY-TYPE (24)
*Jun  4 18:07:01.620: TCP2: Telnet received WONT WINDOW-SIZE (31)
Sw1(config)#
*Jun  4 18:07:03.676: %SYS-5-CONFIG_I: Configured from console by cisco on vty0 (192.168.52.144)
Sw1(config)#
```

EX2: Configuration von SSH ,Hier werden die notwendigen Befehle definiert, um SSH auf einem Gerät zu konfigurieren (IP-Domänenname, VTY-Konfiguration, Generierung eines RSA-Schlüssels mit 2048 Bit). Dieses Skript könnte als Vorbereitung für SSH-Automatisierungstools wie Netmiko oder Ansible nützlich sein. Auch hier wird das gleiche Prinzip wie bei den Telnet-Anmeldeinformationen über eine JSON-Datei genutzt.

Beispiel 2: SSH-Konfiguration über Telnet bei Switch sw2 und Router R1

```
host = item['Host']
user = item['User']
if user != '':
    pass
else:
    user=input(f'enter User name for {host}:\n>>')
password = item['Password']
if password != '':
    pass
else:
    password = getpass.getpass(prompt=f'bitte enter user {user} Password :\n >>')

enable_pw = item['Enable']
if enable_pw != '':
    pass
else:
    enable_pw = getpass.getpass(prompt=f'bitte enter Enable Password for Host {host} :\n >>')

tn = telnetlib.Telnet(host) #telnet Erstellung

tn.write(user.encode('ascii') + b'\n') #username
if password:
    tn.read_until(b'Password: ')
    tn.write(password.encode('ascii') + b'\n')#password
tn.write(b"enable\n")
tn.write(enable_pw.encode('ascii') + b'\n')
tn.write(b"conf t\n")
tn.write(b"line vty 0 4 \n") #configure line vty for ssh
tn.write(b"transport input all \n")
tn.write(b"login local \n")
tn.write(b"exit\n")
tn.write(b"ip domain-name cisco \n")  #domain name
tn.write(b"crypto key generate rsa modulus 2048 \n") #Generate  Crypto  RSA key mit 2048 bit
time.sleep(10)
tn.write(b"\n")
tn.write(b"\n")
tn.write(b"\n")
tn.write(b"do wr \n")
time.sleep(3)
tn.write(b'end\n')
tn.write(b'exit\n')
print(tn.read_all().decode('ascii'))
```

```
root@NetworkAutomation-Dunya:~/python_project# python3 telnet_ssh_config.p

Sw2>enable
Password:
Sw2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Sw2(config)#line vty 0 4
Sw2(config-line)#transport input all
Sw2(config-line)#login local
Sw2(config-line)#exit
Sw2(config)#ip domain-name cisco
Sw2(config)#crypto key generate rsa modulus 2048
The name for the keys will be: Sw2.cisco

% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 5 seconds)

Sw2(config)#
Sw2(config)#
Sw2(config)#
Sw2(config)#do wr
Building configuration...
Compressed configuration from 1728 bytes to 1033 bytes[OK]
Sw2(config)#end
Sw2#exit


R1>enable
Password:
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#line vty 0 4
R1(config-line)#transport input all
R1(config-line)#login local
R1(config-line)#exit
R1(config)#ip domain-name cisco
R1(config)#crypto key generate rsa modulus 2048
% You already have RSA keys defined named R1.cisco.
% They will be replaced.

% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

R1(config)#
R1(config)#
R1(config)#
R1(config)#do wr
Building configuration...
[OK]
```

**WITERE INFO& LINKE :
HTTPS://DOCS.PYTHON.ORG/3/LIBRARY/TELNETLIB.HTML
HTTPS://PYNENG.READTHEDOCS.IO/EN/LATEST/BOOK/18_SSH_TELNET/TELNETLIB.HTML

NETMIKO

Netmiko ist ein Python-Bibliothek, die SSH-Verbindungen zu Netzwerkgeräten vereinfacht und dabei hilft, Netzwerkautomatisierungsaufgaben effizient zu erledigen . Netmiko verwendet Paramiko, aber in erleichtere Anwendung Art und erstellt aber auch Schnittstellen und Methoden, die für die Arbeit mit Netzwerkgeräten erforderlich sind ,dabei einiges kürzer als das Paramiko-Pendant .
*Neben der SSH-Unterstützung unterstützt Netmiko auch Secure Copy, Telnet-Verbindungen und serielle Verbindungen.
Diese sind die wichtige Zwecke dieser Bibliothek:

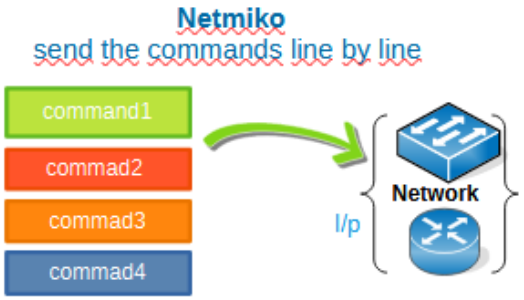
-Stellen Sie erfolgreich eine SSH-Verbindung zum Gerät her.
-Vereinfachen Sie die Ausführung, den Abruf und die Formatierung von Show-Befehlen.
-Vereinfachen Sie die Ausführung von Konfigurationsbefehlen.
-Abstrahieren Sie einen Großteil der einfachen Mechanismen der Interaktion mit Geräten.
-Führen Sie das oben Genannte bei einem breiten Spektrum von Netzwerkanbietern und -plattformen durch.
Pro: Vereinfacht die Interaktion mit Netzwerkgeräten und unterstützt die Unterstützung mehrerer Anbieter.
Con: Beschränkt auf SSH und unterstützt keine anderen Netzwerkverwaltungsprotokolle.

Sub-modules

netmiko.cisco.cisco_asa_ssh
Subclass specific to Cisco ASA.
netmiko.cisco.cisco_ftd_ssh
Subclass specific to Cisco FTD.
netmiko.cisco.cisco_ios
netmiko.cisco.cisco_nxos_ssh
netmiko.cisco.cisco_s200
netmiko.cisco.cisco_s300
netmiko.cisco.cisco_tp_tcce
CiscoTpTcCeSSH Class Class to manage C
Also working for Cisco Expressway/VCS ..
netmiko.cisco.cisco_viptela
Subclass specific to Cisco Viptela.
netmiko.cisco.cisco_wlc_ssh
Netmiko Cisco WLC support.
netmiko.cisco.cisco_xr

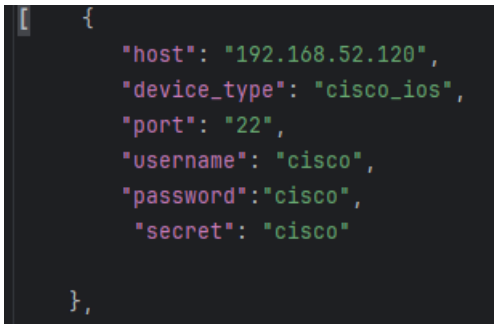
Die main Supported Platform:
Arista vEOS,Cisco IOS.Cisco IOS-XE,Cisco IOS-XR,Cisco NX-OS,Cisco SG300,Juniper Junos,Linux
*gibt es noch mehre Platforms ,die im Test oder Expermint Mode(sehe das Link bitte)
-supported Platform: https://github.com/ktbyers/netmiko/blob/develop/PLATFORMS.md
Die Main Methode supported für meisten Netzwerk Geräte :

- Connect via SSH : ConnectHandler()
-Send Command :
send_command - send one command
send_config_set - send list of commands or command in configuration mode
send_config_from_file - send commands from the file (uses send_config_set method inside)
send_command_timing - send command and wait for the output based on timer
Additional methods :



-Enable Mode: enable()
config_mode - switch to configuration mode: ssh.config_mode
exit_config_mode - exit configuration mode: ssh.exit_config_mode
check_config_mode - check whether netmiko is in configuration mode (returns True if in configuration mode and False if not): ssh.check_config_mode
find_prompt - returns the current prompt of device: ssh.find_prompt
commit - commit on IOS-XR and Juniper: ssh.commit
disconnect - terminate SSH connection

Parameters Dictionary for defining device:



** hier sind die Parameters ,die bei von ConnectHandler Netmiko als Dictionary **kwargs gegben ,zum Erstellung den SSH Connection genutzt werden, **trotzt die Ähnlichkeit mit NAPALMIKO Parameter ,aber gibt es Unterschied mit Benennung ,. So es ist besser für jede Bibliothek eigene Dictionary(or json File,bei mehre Geräte Fall) zu erstellen .

**Exampls und Screenshoot für Einsatz von Netmiko ,finden Sie bei hier Network Manager Programm

INFO&LINKE:
-HTTPS://PYNET.TWB-TECH.COM/BLOG/NETMIKO-PYTHON-LIBRARY.HTML
-EXAMPLES: HTTPS://GITHUB.COM/KTBYERS/NETMIKO/BLOB/DEVELOP/EXAMPLES.MD

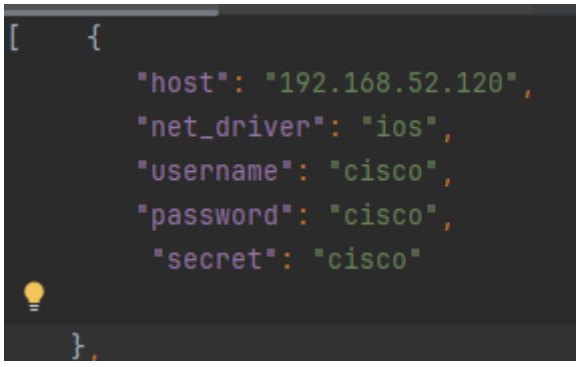
NAPALM (NETWORK AUTOMATION AND PROGRAMMABILITY ABSTRACTION LAYER WITH MULTIVENDOR SUPPORT)

NAPALM ist eine Python-Bibliothek, die eine Reihe von Funktionen implementiert, um über eine einheitliche API mit verschiedenen Netzwerkgeräte-Betriebssystemen zu interagieren.NAPALM unterstützt mehrere Methoden, um eine Verbindung zu den Geräten herzustellen, Konfigurationen zu manipulieren oder Daten abzurufen.Aufgrund seiner Flexibilität kann NAPALM in weit verbreitete Automatisierung Frameworks(wie Ansible,SaltStack..) integriert werden.

Supported Network Operating Systems:

Arista EOS,Cisco IOS,Cisco IOS-XR,Cisco NX-OS,Juniper JunOS

Parameters Dictionary for defining device:



** hier sind die benötige Parameters ,zum Erstellung die Verbindung , über driver (or hier net_drive gennant) entscheidet welche Platform abgerufen soll ,**hier ist cisco IOS .Nachher wird über command open_connection ,bei rufen host IP,User ,Password,enable) ein Verbindung erstellt ,**normalerweise benutzt Napalm API Call ,aber bei Cisco IOS ,wegen fehlende API ,wird über Netmiko den SSH Protocol benutzt.

Supported Methode :

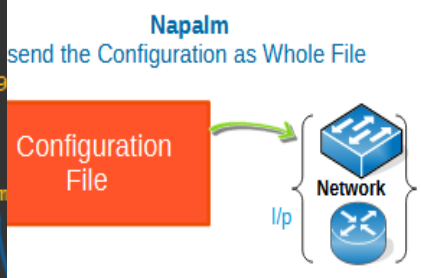
Napalm enthält mehre Methoden,mit dem versucht, eine gemeinsame Schnittstelle und Mechanismen bereitzustellen, um die Konfiguration voranzutreiben und Status daten von Netzwerkgeräten abzurufen über gebaute Template .

Configuration support matrix

	EOS	IOS	IOSXR	IOSXR_NETCONF	JUNOS	NXOS	NXOS_SSH
get_arp_table	✓	✓	✗	✗	✗	✗	✓
get_bgp_config	✓	✓	✓	✓	✓	✗	✗
get_bgp_neighbors	✓	✓	✓	✓	✓	✓	✓
get_bgp_neighbors_detail	✓	✓	✓	✓	✓	✗	✗
get_config	✓	✓	✓	✗	✓	✓	✓
get_environment	✓	✓	✓	✓	✓	✓	✓
get_facts	✓	✓	✓	✓	✓	✓	✓
get_firewall_policies	✗	✗	✗	✗	✗	✗	✗
get_interfaces	✓	✓	✓	✓	✓	✓	✓
get_interfaces_counters	✓	✓	✓	✓	✓	✗	✓
get_interfaces_ip	✓	✓	✓	✓	✓	✓	✓
get_ipv6_neighbors_table	✗	✓	✗	✗	✓	✗	✗
get_lldp_neighbors	✓	✓	✓	✓	✓	✓	✓
get_lldp_neighbors_detail	✓	✓	✓	✓	✓	✓	✓
get_mac_address_table	✓	✓	✓	✓	✓	✓	✓
get_network_instances	✓	✓	✗	✗	✓	✓	✓
get_ntp_peers	✗	✓	✓	✓	✓	✓	✓
get_ntp_servers	✓	✓	✓	✓	✓	✓	✓
get_ntp_stats	✓	✓	✓	✓	✓	✓	✗
get_optics	✓	✓	✗	✗	✓	✗	✓
get_probes_config	✗	✓	✓	✓	✓	✗	✗
get_probes_results	✗	✗	✓	✓	✓	✗	✗
get_route_to	✓	✗	✗	✗	✗	✗	✗
get_snmp_information	✓	✓	✓	✓	✓	✓	✓
get_users	✓	✓	✓	✓	✓	✓	✓
get_vlans	✓	✓	✗	✗	✓	✓	✓
is_alive	✓	✓	✓	✓	✓	✓	✓
ping	✓	✓	✗	✗	✓	✓	✓
traceroute	✓	✓	✓	✓	✓	✓	✓

Anders als Netmiko, das die Konfiguration Zeile für Zeile schickt, sendet Napalm die Konfiguration als Gesamteinheit (Datei) an das Gerät. Dies macht die Fehlersuche etwas schwieriger und erfordert Vorsicht und Genauigkeit bei der Erstellung der Konfigurationsdatei, da jeder Befehl zum System passen muss. Andernfalls tritt ein Ausnahmefehler auf. Allerdings bietet dieses Konzept in Napalm (Speicherung der Konfiguration als einheitliche Datei) Funktionen und Möglichkeiten wie **Rollab Compare ,Replace ,und Committ der Konfiguration, was mit Netmiko nicht möglich ist.

```
v3#dir
irectory of flash0:/

  1  drw-           0  Jan 30 2013 00:00:00 +00:00  boot
264  drw-           0  Oct 14 2013 00:00:00 +00:00  config
267  -rw-   118966680  Apr 23 2019 00:00:00 +00:00  vios_l2-adventerprisek9
268  -rw-      524288  Jun 20 2024 11:55:50 +00:00  nvram
269  -rw-         325  Jun 20 2024 11:58:14 +00:00  e1000_bia.txt
270  -rw-          35  Jun 20 2024 12:00:50 +00:00  pnp-tech-time
271  -rw-       32400  Jun 20 2024 12:00:52 +00:00  pnp-tech-discovery-summ
272  -rw-          29  Jun 20 2024 12:26:38 +00:00  merge_config.txt
273  -rw-        3599  Jun 20 2024 12:23:46 +00:00  candidate_config.txt
274  -rw-        3655  Jun 20 2024 12:27:16 +00:00  rollback_config.txt
```



**Napalm bietet ein hervorragendes get_Information-Feature, das die Automatisierung von Assurance-Funktionen und die Übersicht über das gesamte Netzwerk ermöglicht. Dieses

Feature stellt eine Alternative zu den üblichen show-Befehlen dar, die auf jedem einzelnen Gerät ausgeführt werden müssen. Damit werden wichtige Informationen zur Fehlerbehebung für die gesamte Netzwerkstruktur gesammelt. Zusätzlich sind integrierte Funktionen wie Traceroute und Ping enthalten, die die Netzwerkverbindung und Routing-Prüfung ermöglichen.

Hier ist z,b 3 Screenschoots von **Get-Facts**(information über system,os_version,interface, uptime) und **Get-Arp** ,**Get-Interfaces_Counter**. **Output wurde als TextFSM- Dictionary Sortiet ,das hilft bei weiter Processing auf den Information als Json file ,und für besser Ansicht

```
root@NetworkAutomation-Dunya:~/python_project# python3 napa_test.py
connecting to Host192.168.52.100
{
    "fqdn": "R1.cisco",
    "hostname": "R1",
    "interface_list": [
        "Ethernet0/0",
        "Ethernet0/1",
        "Ethernet0/2",
        "Ethernet0/3",
        "Ethernet1/0",
        "Ethernet1/1",
        "Ethernet1/2",
        "Ethernet1/3",
        "Serial2/0",
        "Serial2/1",
        "Serial2/2",
        "Serial2/3",
        "Serial3/0",
        "Serial3/1",
        "Serial3/2",
        "Serial3/3",
        "Loopback1"
    ],
    "model": "Unknown",
    "os_version": "Linux Software (I86BI_LINUX-ADVENTERPRISEK9-M), Version 15.5(2)T, DEVELOPMENT TEST SOFTWARE"
    "serial_number": "2048001",
    "uptime": 3480.0,
    "vendor": "Cisco"
}
disconnecting from Host192.168.52.100
connecting to Host192.168.52.120
{
    "fqdn": "Sw1.cisco",
    "hostname": "Sw1",
    "interface_list": [
        "Ethernet0/0",
        "Ethernet0/1",
        "Ethernet0/2",
        "Ethernet0/3",
        "Ethernet1/0",
        "Ethernet1/1",
        "Ethernet1/2",
        "Ethernet1/3",
        "Ethernet2/0",
        "Ethernet2/1",
        "Ethernet2/2",
        "Ethernet2/3",
        "Ethernet3/0",
        "Ethernet3/1",
        "Ethernet3/2",
        "Ethernet3/3",
        "Vlan1"
    ],
    "model": "Unknown",
    "os_version": "Solaris Software (I86BI_LINUX
    "serial_number": "2048003",
    "uptime": 3480.0,
    "vendor": "Cisco"
}
disconnecting from Host192.168.52.120
root@NetworkAutomation-Dunya:~/python_project#
```

```
root@NetworkAutomation-Dunya:~/python_project# nano napa_test.py
root@NetworkAutomation-Dunya:~/python_project# python3 napa_test.py
connecting to Host192.168.52.100
{'interface': 'Ethernet0/0', 'mac': 'AA:BB:CC:00:01:00', 'ip': '192.168.52.100', 'age': -1.
{'interface': 'Ethernet0/0', 'mac': 'A6:6A:25:8E:55:88', 'ip': '192.168.52.192', 'age': 6.0
disconnecting from Host192.168.52.100
connecting to Host192.168.52.120
{'interface': 'Vlan1', 'mac': 'AA:BB:CC:80:03:00', 'ip': '192.168.52.120', 'age': -1.0}
{'interface': 'Vlan1', 'mac': 'A6:6A:25:8E:55:88', 'ip': '192.168.52.192', 'age': 6.0}
disconnecting from Host192.168.52.120
root@NetworkAutomation-Dunya:~/python_project# nano napa_test.py
root@NetworkAutomation-Dunya:~/python_project# python3 napa_test.py
connecting to Host192.168.52.100
```

```
    "Ethernet0/2": {
        "rx_broadcast_packets": 0,
        "rx_discards": 0,
        "rx_errors": 0,
        "rx_multicast_packets": 0,
        "rx_octets": 0,
        "rx_unicast_packets": 0,
        "tx_broadcast_packets": -1,
        "tx_discards": 0,
        "tx_errors": 0,
        "tx_multicast_packets": -1,
        "tx_octets": 0,
        "tx_unicast_packets": 0
    },
    "Ethernet0/3": {
        "rx_broadcast_packets": 0,
        "rx_discards": 0,
        "rx_errors": 0,
        "rx_multicast_packets": 0,
        "rx_octets": 0,
        "rx_unicast_packets": 0,
        "tx_broadcast_packets": -1,
        "tx_discards": 0,
        "tx_errors": 0,
        "tx_multicast_packets": -1,
        "tx_octets": 0,
        "tx_unicast_packets": 0
    },
    "Ethernet1/0": {
```

INFO &LINKE :
HTTPS://NAPALM.READTHEDOCS.IO/EN/LATEST/
HTTPS://WWW.PACKETSWITCH.CO.UK/NAPALM-NETWORK-AUTOMATION/

**Weitere Beispiele und Screenshoot für Einsatz von Napalm ,finden Sie bei Network Manager Programm Section

LINUX KONFIGURATION BEISPIEL(USING PARAMIKO-SSHV2

Paramiko ist eine Python-Implementierung von SSHv2, die sowohl client- als auch serverseitige Funktionalitäten bereitstellt. Es wird häufig zum Ausführen von Befehlen auf Remote-Systemen verwendet. Mit der Bibliothek können Nutzer Befehle senden, die sie normalerweise manuell eingeben würden, und die Ergebnisse jeder Befehlsausführung parsen, was auch als Screen Scraping bezeichnet wird.
Pro: Automatisiert und verwaltet SSH-Verbindungen und unterstützt die Schlüsselauthentifizierung.
Contra: Relativ geringere Leistung im Vergleich zu einigen anderen Bibliotheken wie libssh.

Netmiko bietet im Vergleich zu Paramiko bei der Netzwerkprogrammierung einfachere Programmierinterfaces. Trotzdem spielt Paramiko eine wichtige Rolle, beispielsweise bei der Verwaltung von Linux-Systemen (neben anderen Protokollen).
EX*Hier habe ich ein kleine Automatisierung Aufgabe (add User mit Home DNur als kleine Präsentation, was man noch in diesem Bereich automatisieren kann (z. B. Patch-Management und Software-Installation, Datei- und Verzeichnisverwaltung, Benutzer- und Gruppenverwaltung, Berechtigungen, Backup etc.), zeigt dieses kleine Skript, wie man unter Linux Benutzer hinzufügt und überprüft, ob sie korrekt erstellt wurden.irectory) für List of Server(im Toplogy Dibian 1&Dibian2) .
* Als kleine Präsentation ,was kann Man noch eben in dem Bereich automatisieren(z.b Patch Management &Software Installation ,File Und Directory Management,User& Group &Permissions,Backup..etc) ,mit Hilfe diese kleine Skript

```python
import paramiko
import time

ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())


linux = [{'hostname': '192.168.52.160', 'port': '22', 'username':'debian', 'password': 'debian'},{'hostname': '192.168.52.1>
for item in linux:
        print(f'Connecting to {item["hostname"]}')
        ssh_client.connect(**item, look_for_keys=False, allow_agent=False)
        new_user=input(f'enter den neue username  for Host {item["hostname"]} \n >>>  ')


        stdin, stdout, stderr = ssh_client.exec_command(f'sudo useradd -m -g users {new_user}\n', get_pty=True)

        stdin.write('debian\n ') # this is the sudo password
        time.sleep(2) # waiting for the remote server to finish

        stdin, stdout, stderr = ssh_client.exec_command('cat /etc/passwd\n')
        print(stdout.read().decode())
        time.sleep(1)

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

Beispiel-Skript:
Linux-Befehle zum Hinzufügen eines Benutzers mit Home-Verzeichnis und zur Gruppe users hinzufügen:, brows

den User datei /etc/passwd ,um sicher zu sein ,dass den nuen User würde richtig hinzugefügt wurde (client bei client)

1-sudo useradd -m -g users {new_user}

2-cat /etc/passwd

```
root@NetworkAutomation-1:~/python_project# nano linux.py
root@NetworkAutomation-1:~/python_project# python3 linux.py
Connecting to 192.168.52.160
enter den neue username  for Host 192.168.52.160
 >>>  dunya1
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
uuidd:x:100:105::/run/uuidd:/usr/sbin/nologin
messagebus:x:101:106::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:996:996:systemd Resolver:/:/usr/sbin/nologin
tcpdump:x:102:107::/nonexistent:/usr/sbin/nologin
sshd:x:103:65534::/run/sshd:/usr/sbin/nologin
polkitd:x:995:995:polkit:/nonexistent:/usr/sbin/nologin
debian:x:1000:1000:Debian:/home/debian:/bin/bash
gns3:x:1001:1001:GNS3,,,:/home/gns3:/bin/bash
u2:x:1002:1002::/home/u2:/bin/sh
u3:x:1003:1003::/home/u3:/bin/sh
{new_user}:x:1004:100::/home/{new_user}:/bin/sh
dunya:x:1005:100::/home/dunya:/bin/sh
duny:x:1006:100::/home/duny:/bin/sh
dunya1:x:1007:100::/home/dunya1:/bin/sh

Closing connection
Connecting to 192.168.52.170
enter den neue username  for Host 192.168.52.170
 >>>  dunya1
```

Weitere Automatisierungsmöglichkeiten:

1-Patch-Management & Software-Installation: Automatisierung der Installation und Aktualisierung von Softwarepaketen.
2-Datei- und Verzeichnisverwaltung: Automatisierung der Erstellung, Löschung und Verwaltung von Dateien und Verzeichnissen.
3-Benutzer- und Gruppenverwaltung: Automatisierung der Erstellung und Verwaltung von Benutzern und Gruppen sowie deren Berechtigungen.
4-Backup: Automatisierung der Datensicherung und -wiederherstellung.
Durch solche Skripte kann die Effizienz und Zuverlässigkeit der Systemverwaltung erheblich gesteigert werden.

*Bash äquivalent Kommando mit mehr Option zu prüfen der Erstellung der User &Home Directory(hier handelt sich von nicht selbst geschriebene bash skript,sondern von netz)

bash — Code kopieren

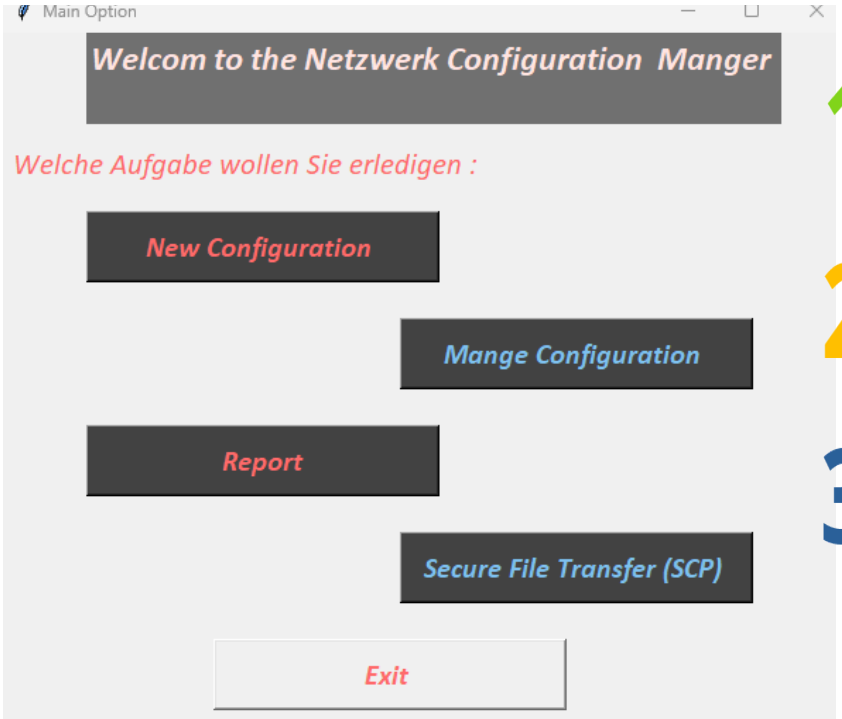
```bash
#!/bin/bash

# Benutzername und Gruppe
USERNAME="neuer_benutzer"
GROUP="users"

# Benutzer mit Home-Verzeichnis hinzufügen und zur Gruppe hinzufügen
sudo useradd -m -G $GROUP $USERNAME

# Überprüfen, ob der Benutzer erfolgreich hinzugefügt wurde
if id "$USERNAME" &>/dev/null; then
    echo "Benutzer $USERNAME wurde erfolgreich hinzugefügt."
else
    echo "Fehler: Benutzer $USERNAME wurde nicht hinzugefügt."
    exit 1
fi

# Durchsuchen der Datei /etc/passwd nach dem Benutzer
if grep -q "^$USERNAME:" /etc/passwd; then
    echo "Benutzer $USERNAME ist in der Datei /etc/passwd vorhanden."
else
    echo "Fehler: Benutzer $USERNAME ist nicht in der Datei /etc/passwd vorhanden."
    exit 1
fi
```

NETZWERK KONFIGURATION MANAGER PROGRAMM

Introduktion :

Der Network Configuration Manager bietet über eine benutzerfreundliche GUI den Zugang zu Netzwerkgeräten mittels Netmiko und Napalm.
Die Authentifikation Informationen (Management-IP (Host), Benutzername, Passwort und Enable-Passwort) werden in einer JSON-Datei gespeichert. Das System ermöglicht flexible Speicheroptionen für diese Datei. Sollte ein Sicherheitsrisiko bestehen und der Administrator die kritischen Informationen nicht in der Datei speichern wollen, prüft das Programm das Vorhanden json Datei. Falls die username oder password fehlen, kann der Administrator die Informationen manuell eingeben.

*Die Plattform oder der Gerätetyp kann über eine JSON-Datei angegeben werden, sei es Cisco (IOS, IOS-XR, NX-OS), Arista, Juniper oder andere Typen (Switch, Router, Firewall, ...). Diese Informationen werden in der JSON-Datei gespeichert und dem Programm zugeführt. Bei Netmiko erfolgt dies über den Key „network-driver“, bei Napalm über den Key „net-driver“.



1 • Read the Json File with Device List &Login info

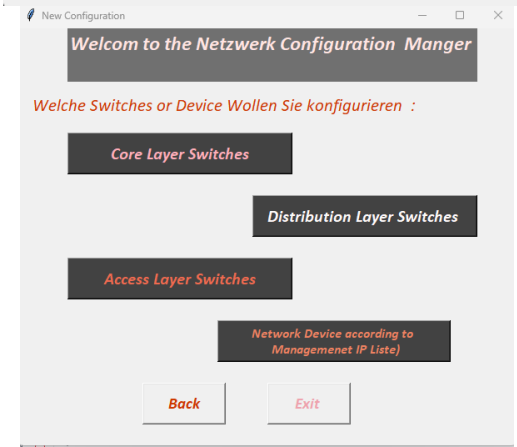
2 • Establishe Connection to each Device in json List

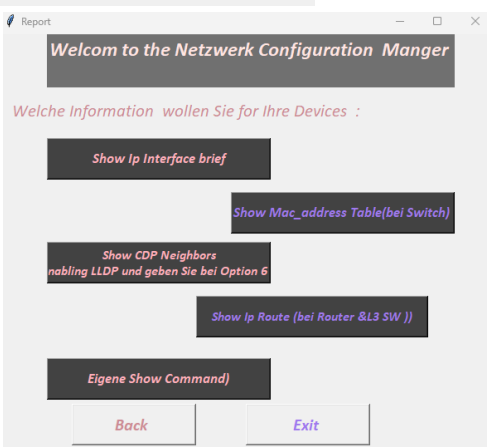
3 • Send the Authenticate Information and enter enable Mode

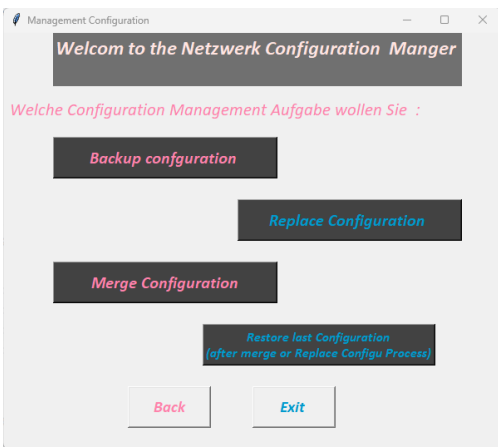
4 • With **New & Mange** Configuration ,read the Configuration File and apply to the Device

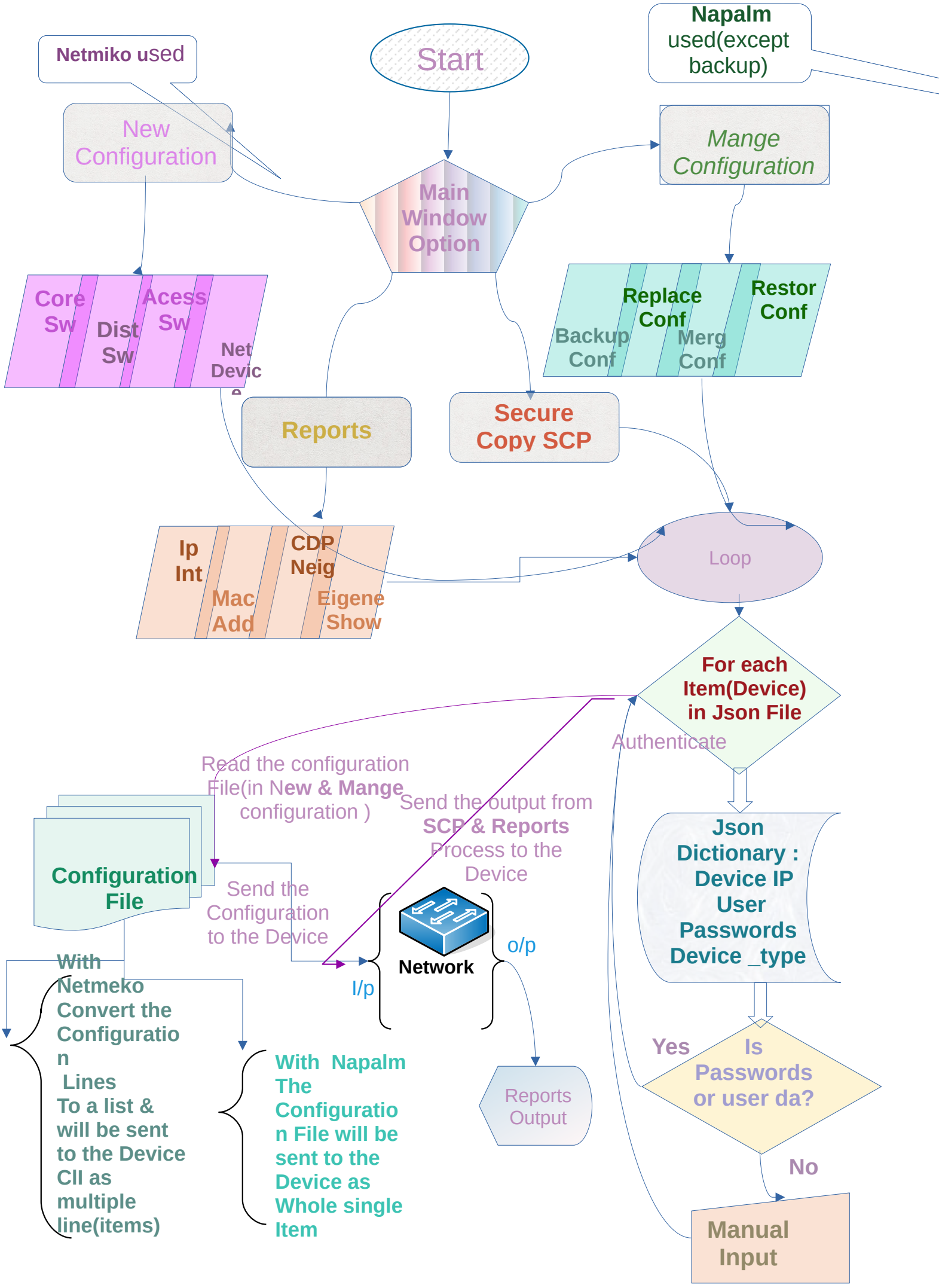
5 • In **SCP** connect to the Device and copy Files

6 In **Reports** ,take the Output from show command to File and the screen

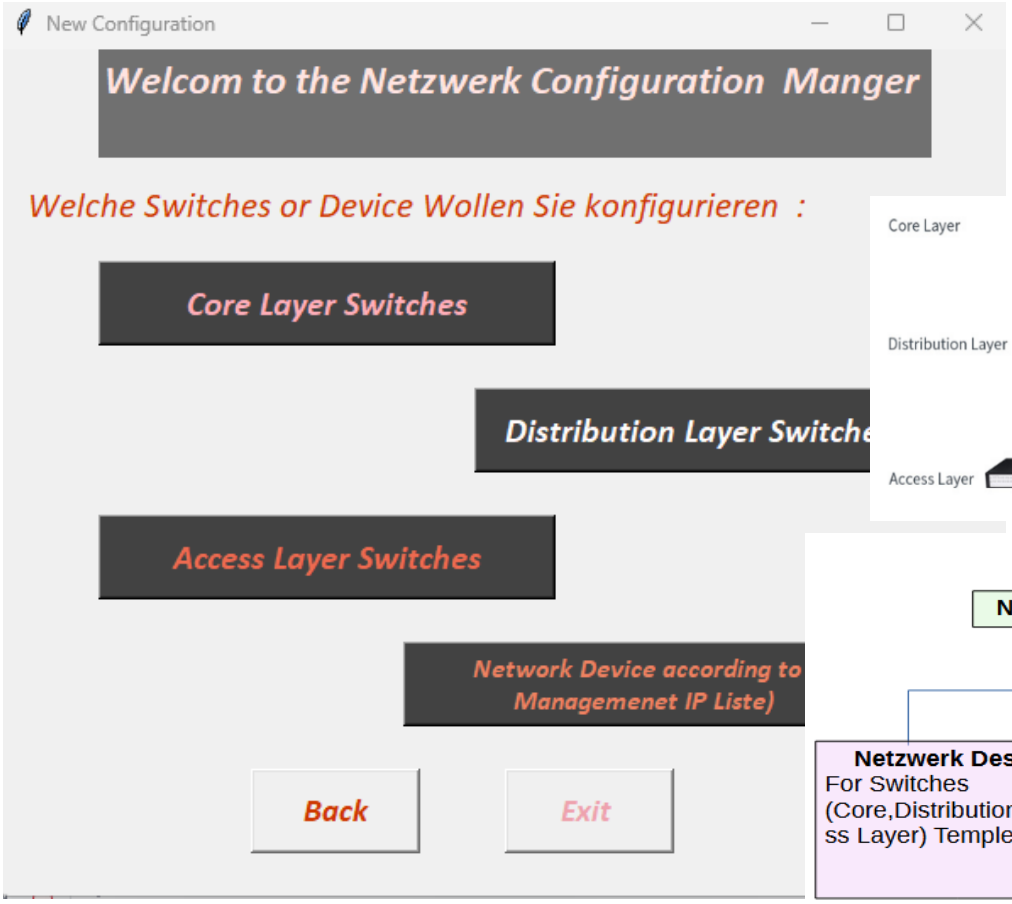


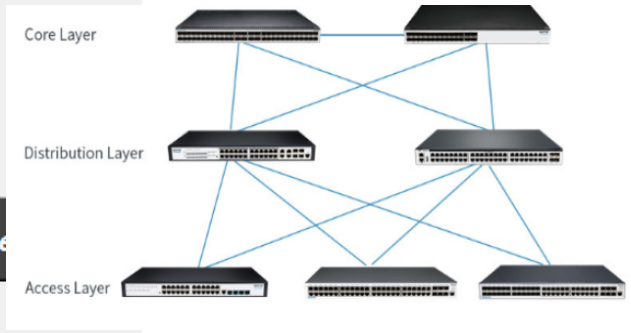


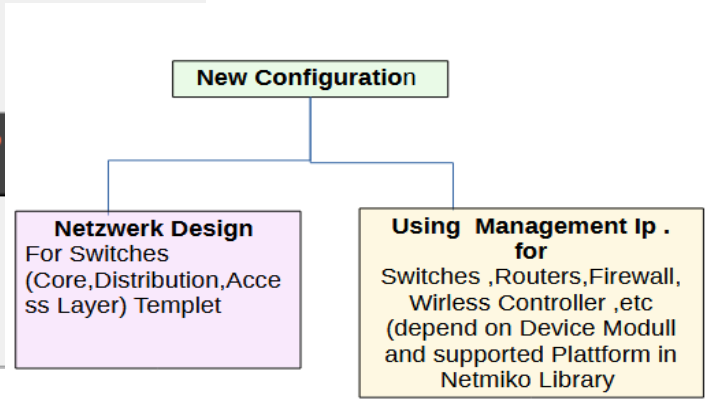



Netmiko used
Start
Napalm used(except backup)
New Configuration
Mange Configuration
Main Window Option
Core Sw
Dist Sw
Acess Sw
Net Device
Backup Conf
Replace Conf
Merg Conf
Restor Conf
Reports
Secure Copy SCP
Ip Int
Mac Add
CDP Neig
Eigene Show
Loop
For each Item(Device) in Json File
Authenticate
Read the configuration File(in New & Mange configuration)
Send the output from SCP & Reports Process to the Device
Configuration File
Send the Configuration to the Device
Network
I/p
o/p
Json Dictionary : Device IP User Passwords Device _type
Yes
Is Passwords or user da?
No
Manual Input
With Netmeko Convert the Configuration Lines To a list & will be sent to the Device Cll as multiple line(items)
With Napalm The Configuration File will be sent to the Device as Whole single Item
Reports Output

NEW KONFIGURATION







Bei der neuen Konfigurationsauswahl habe ich Netmiko als Haupt-Connector verwendet, weil Netmiko Unterstützung für eine größere Auswahl an Plattformen bietet und weniger Fehler verursacht. Dies liegt daran, dass nur die Befehle abgelehnt werden, die nicht zum System passen, und nicht das gesamte Programm gestoppt wird, wie es bei Napalm der Fall ist (siehe Abschnitt "Netmiko & Napalm" für Einschränkungen beider Bibliotheken).

Grundsätzlich übersetzt das Programm die Konfigurationsdatei in eine Liste und überträgt diese dann Punkt für Punkt (Zeile für Zeile) über den Netmiko SSH-Connector an das jeweilige Gerät.

Configuration Prinzip im Programm:

Das vorgestellte Konfigurationsprinzip wird einmal für Switches über das 3-Layer-Netzwerkdesign (Core-, Distribution-, Access-Layer-Template) oder über die Management-IP angewendet (für allgemeine Netzwerkgeräte, die von der Netmiko-Bibliothek unterstützt werden).

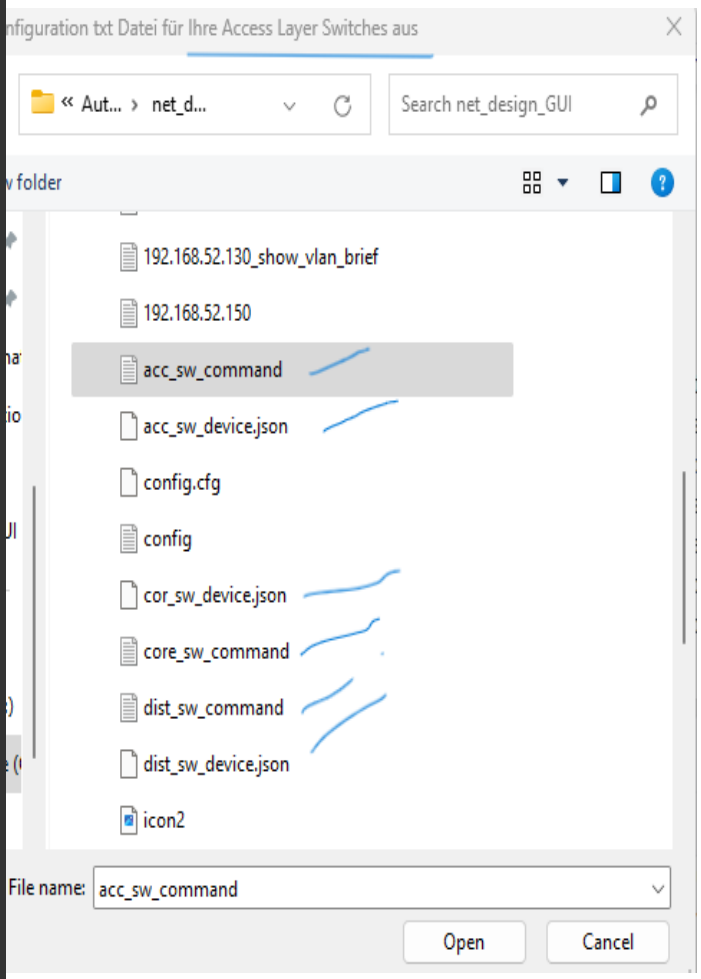
1-Beim ersten Prinzip, dem Netzwerk-Layer, wird eine allgemeine Konfigurationsdatei erstellt, die je nach Rolle des Switches (Core, Distribution, Access Layer) die Hauptkonfiguration für diese Layer-Switches enthält.,wie algemein Policy ACL ,AAA ,Security ,Assurance etc,sehen sie unten Bild ,als Beispiel für solche File)und auf alle Switches diese Layer verteilt .
*Falls der Admin diese Konzepte fortsetzen möchte, muss er einmal die Geräte in 3 JSON-Datei sortieren und drei verschiedene Typen von Konfigurationsdateien erstellen. Alternativ

kann der Admin auf dieses Prinzip verzichten und jedem Gerät eine Konfigurationsdatei mit den gesamten Einstellungen zuweisen(Management IP)
2-Für das zweite Prinzip (Management-IP):
Hier wird eine zusätzliche Konfigurationsdatei für jeden Switch erstellt und über die Management-IP-Liste verteilt. Dies ist besonders nützlich für spezielle Konfigurationen, wie zum Beispiel Interface-Einstellungen.

*Beispiel von Allgemein Configuration File

```
  GNU nano 4.8
vtp mode transparent
spanning-tree mode rapid-pvst
udld enable
errdisable recovery cause all
port-channel load-balance src-dst-ip

no ip http server
no ip http secure-server

snmp-server community test RO
snmp-server community test RW

vlan 100
 name Data
vlan 101
 name Voice
vlan 102
 name Test
interface vlan 1
 description In-band Management

ip dhcp snooping vlan 100,101

no ip dhcp snooping information option
ip dhcp snooping
ip arp inspection vlan 100,101
spanning-tree portfast bpduguard default

interface range GigabitEthernet 2/0 - 3
 switchport
 switchport access vlan 100
 switchport voice vlan 101
 switchport host
 switchport port-security maximum 2
 switchport port-security
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security violation restrict
 ip arp inspection limit rate 100
 ip dhcp snooping limit rate 100
```

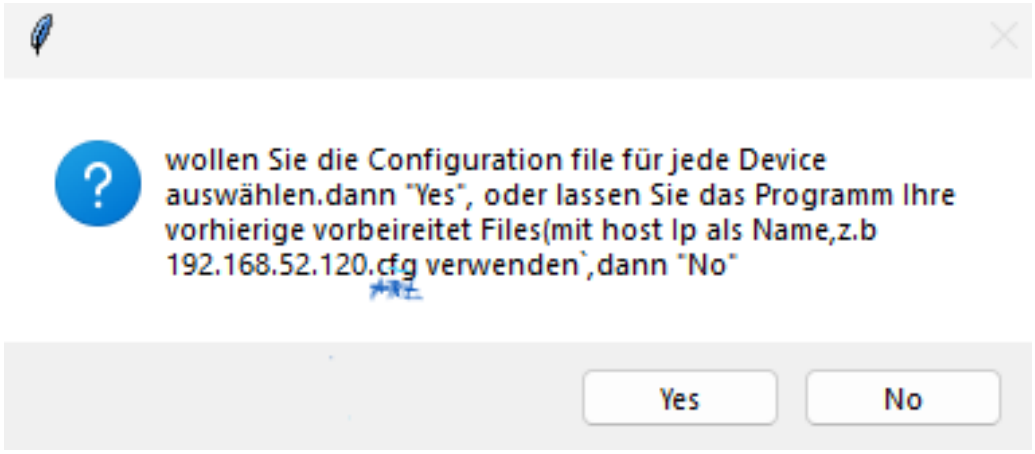
*open File Dialog&Config File Selection(access-switch)



**Es ist ideal, beide Prinzipien zu kombinieren. Das heißt, im ersten Schritt die Konfiguration nach der Rolle des Switches(network Design) zu verteilen und dann im zweiten Schritt spezielle Einstellungen für jeden Switch(Management_IP) vorzunehmen,das heißt ,4 verschiedene Configuration File Typen zu erstellen . Dies kann jedoch komplex sein und erfordert viel Arbeitszeit.

**Bei andere Devices ,wie Routers,Firwall ,Wirlless Controller ..etc könnte die Configuration direkt über Management IP ,nach definieren Device Type in json File , verteilt werden .

Configuration File Auswahl :

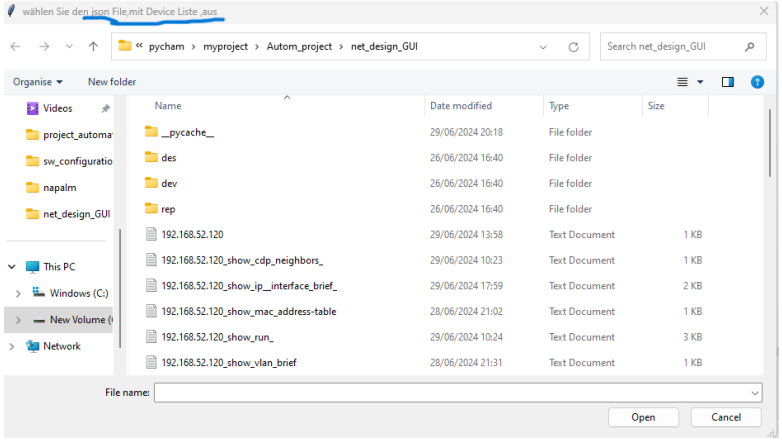


Das Programm fragt die Administratorin oder den Administrator, ob die Konfigurationsdatei manuell eingegeben werden soll oder ob das Programm (über die JSON-Datei mit den Host-IPs) die passende Konfigurationsdatei automatisch finden soll(managementIp.txt z.b 192.168.52.120.txt). Falls die Antwort "Ja" lautet, öffnet sich ein Dialogfenster, um die Konfigurationsdatei auszuwählen. Bei "Nein" läuft ein Loop, in dem das Programm für jedes Gerät über die Host-IP in der JSON-Liste die entsprechende Konfigurationsdatei sucht.

```python
def list_Sw():  # for Switch from list with Management IP
    json_file = filedialog.askopenfilename(title='wählen Sie den json File,mit Device Liste ,aus', initialdir='.')
    with open(json_file, 'r') as f:
        dev_list = json.load(f)
    for item in dev_list:
        host = item['host']
        print(f'connecting to Host {host}  ')
        ask = messagebox.askquestion(message='wollen Sie die Configuration file für jede Device auswählen.dann "Yes", oder las
        if ask =='yes':
            filename = filedialog.askopenfilename(title='wählen Sie den Configuration File aus')#,filetypes='*.*'
        elif ask == 'no':
            filename= f'{host}.txt'
        with open(filename) as f:
            list_lines = f.read().splitlines()  ## read the command as list
            print(list_lines)
        execute(item, list_lines)
    quit()
```

Netmiko supprt wide Variante von Plattform,und bei jede Device Model ,mehre Methode ,so Kann man mit kleine Änderung auf den geschrieben Code ,für ändere Model, Automatisierung Aufgabe beinhalten : Beispiel Hier Methode bei Cisco ASA FTD,WLC ,oder andere Herstelle wie Arista ,Cisco NX-OS,,Juniper

*open Json File Dialog



*configuration Beispiel von SW1 & SW2 in Topologie

```
Sw2#
Closing connection
connecting t Host 192.168.52.120
['vtp mode transparent', 'spanning-tree mode rapid-pvst', 'errdisable recovery cause all
t-ip', '', 'no ip http server', '', 'snmp-server community test RO', 'snmp-server commun
 Data', 'vlan 101', ' name Voice', 'vlan 102', ' name Test ', 'interface vlan 1', ' desc
 dhcp snooping vlan 100,101', 'no ip dhcp snooping information option ', 'ip dhcp snoopi
uguard default', '', 'interface range e 2/0 - 3', ' switchport', ' switchport access vla
, ' switchport host', ' switchport port-security maximum 2 ', ' switchport port-security
time 2', ' switchport port-security aging type inactivity ', ' switchport port-security
ping limit rate 100', ' ip verify source', ' exit']
SSH connection established to 192.168.52.120:22
Interactive SSH session established
vtp mode transparent
Device mode already VTP Transparent for VLANS.
Sw1(config)#spanning-tree mode rapid-pvst
Sw1(config)#errdisable recovery cause all
Sw1(config)#port-channel load-balance src-dst-ip
Sw1(config)#
Sw1(config)#no ip http server
Sw1(config)#
Sw1(config)#snmp-server community test RO
Sw1(config)#snmp-server community test RW
Sw1(config)#
Sw1(config)#vlan 100
Sw1(config-vlan)# name Data
Sw1(config-vlan)#vlan 101
Sw1(config-vlan)# name Voice
Sw1(config-vlan)#vlan 102
Sw1(config-vlan)# name Test
Sw1(config-vlan)#interface vlan 1
Sw1(config-if)# description In-band Manage
Sw1(config-if)#
Sw1(config-if)#ip dhcp snooping vlan 100,1
Sw1(config)#no ip dhcp snooping informatic
Sw1(config)#ip dhcp snooping
Sw1(config)#
Sw1(config)#spanning-tree portfast bpdugua
Sw1(config)#
Sw1(config)#interface range e 2/0 - 3
Sw1(config-if-range)# switchport
Sw1(config-if-range)# switchport access vl
Sw1(config-if-range)# switchport voice vla
Sw1(config-if-range)# switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled

Sw1(config-if-range)# switchport port-secu
Sw1(config-if-range)# switchport port-secu
Sw1(config-if-range)# switchport port-secu
Sw1(config-if-range)# switchport port-secu
Sw1(config-if-range)# switchport port-secu
Sw1(config-if-range)# ip dhcp snooping lim
Sw1(config-if-range)# ip verify source
Sw1(config-if-range)# exit
Sw1(config)#end
Sw1#
Closing connection
root@NetworkAutomation-Dunya:~/python_proj
```

```
Sw1(config-if-range)# switchport port-security violation restrict
Sw1(config-if-range)# ip dhcp snooping limit rate 100
Sw1(config-if-range)# ip verify source
Sw1(config-if-range)# exit
Sw1(config)#end
Sw1#
connecting to Host 192.168.52.130
['vtp mode transparent', 'spanning-tree mode rapid-pvst', 'errdisable recovery cause all', 'port-c
ame Voice', 'vlan 102', ' name Test ', 'interface vlan 1', ' description In-band Management', '',
3', ' switchport', ' switchport access vlan 100', ' switchport voice vlan 101', ' switchport host'
port port-security violation restrict ', ' ip dhcp snooping limit rate 100', ' ip verify source',
SSH connection established to 192.168.52.130:22
Interactive SSH session established
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Sw2(config)#vtp mode transparent
Device mode already VTP Transparent for VLANS.
Sw2(config)#spanning-tree mode rapid-pvst
Sw2(config)#errdisable recovery cause all
Sw2(config)#port-channel load-balance src-dst-ip
Sw2(config)#
Sw2(config)#no ip http server
Sw2(config)#
Sw2(config)#snmp-server community test RO
Sw2(config)#snmp-server community test RW
Sw2(config)#
Sw2(config)#vlan 100
Sw2(config-vlan)# name Data
Sw2(config-vlan)#vlan 101
Sw2(config-vlan)# name Voice
Sw2(config-vlan)#vlan 102
Sw2(config-vlan)# name Test
Sw2(config-vlan)#interface vlan 1
Sw2(config-if)# description In-band Management
Sw2(config-if)#
Sw2(config-if)#ip dhcp snooping vlan 100,101
Sw2(config)#no ip dhcp snooping information option
Sw2(config)#ip dhcp snooping
Sw2(config)#
Sw2(config)#spanning-tree portfast bpduguard default
Sw2(config)#
Sw2(config)#interface range e 2/0 - 3
Sw2(config-if-range)# switchport
Sw2(config-if-range)# switchport access vlan 100
Sw2(config-if-range)# switchport voice vlan 101
Sw2(config-if-range)# switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled

Sw2(config-if-range)# switchport port-security maximum 2
Sw2(config-if-range)# switchport port-security
Sw2(config-if-range)# switchport port-security aging time 2
Sw2(config-if-range)# switchport port-security aging type inactivity
Sw2(config-if-range)# switchport port-security violation restrict
Sw2(config-if-range)# ip dhcp snooping limit rate 100
Sw2(config-if-range)# ip verify source
Sw2(config-if-range)# exit
Sw2(config)#end
Sw2#
root@NetworkAutomation-Dunya:~/python_project#
```

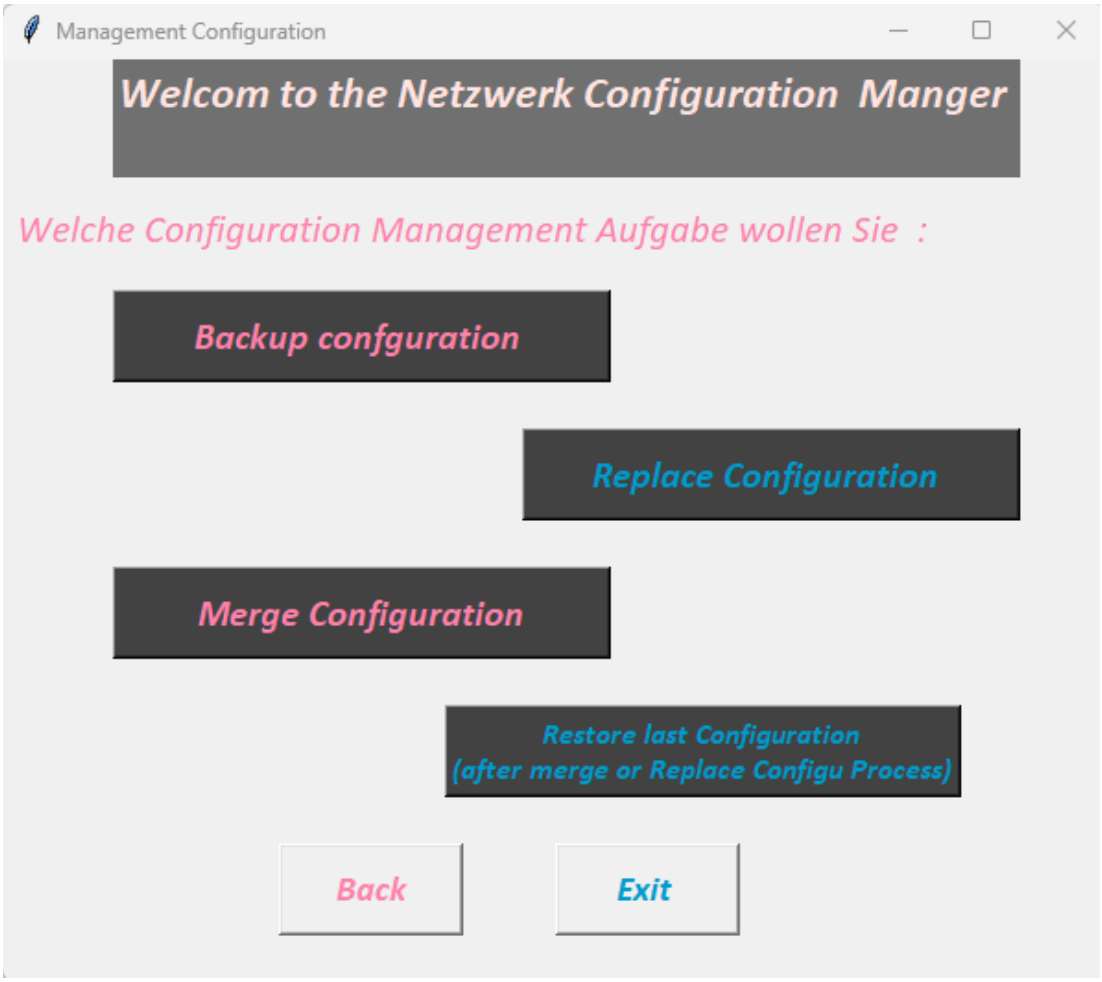
***Python Script**

hier ist non-GUI Form ,in dem habe ich die File Name direkt im Program genannt ,bei GUI Form habe ich statt bestimmte Name Open File -Dialog Box erstzetz

```
        core_sw_list = json.load(f)
    for item in core_sw_list:
        host = item['host']
        print(f'connecting to Host {host}  ')
        execute(item,co_lines)


1 usage
def dist_Sw(): # for Distribution Layer S
    with open('dist_sw_command.txt') as f
        dis_lines = f.read().splitlines()
    print(dis_lines)
    with open('dist_sw_device.json', 'r')
        dist_sw_list = json.load(f)
    for item in dist_sw_list:
        host = item['host']
        print(f'connecting to Host {host}
        execute(item, dis_lines)


1 usage
def acc_Sw(): #for Acess Layer Switch
    with open('acc_sw_command.txt') as f:
        acc_lines = f.read().splitlines()# read the command as list
    print(acc_lines)
    with open('acc_sw_device.json', 'r') as f:
        acc_sw_list = json.load(f)
    for item in acc_sw_list:
        host = item['host']
        print(f'connecting to Host {host}  ')
        execute(item,acc_lines)


1 usage
def list_Sw(): #for Switch from list with Management IP
    with open('list_device_net.json', 'r') as f:
        dev_list = json.load(f)
    for item in dev_list :
        host=item['host']
        print(f'connecting to Host {host}  ')
        with open(f'{host}.txt') as f:
            list_lines = f.read().splitlines()## read the command as l
            print(list_lines)
```

```
from netmiko import ConnectHandler
import json
import time
from netmiko import file_transfer

4 usages
def execute(device, command_list):#open ssh connection and execute command
    try:
        net_connect = ConnectHandler(**device)
        net_connect.enable() # entering the enable mode
        output = net_connect.send_config_set(command_list)
        print(output)
        print(f'connection to Host{host} will be disconnect')
        net_connect.disconnect()
    except:
        print(f'\n the Host {device} is DOWN!!\n')
7 usages
```

MANGE CONFIGRATION



BACKUP CONFIGRATION

Hier wird eine Kopie der Running-Config-Datei mit Datum gespeichert. Diese kann bei Bedarf zur Ersetzung der Konfiguration genutzt werden, falls eine ältere Konfigurationsversion wiederhergestellt werden soll.
*Bei Nutzung Replace Configuration Choice als Option für Wiederherstellung die Configuration ,bitte achten Sie darauf, dass die Konfigurationsdatei mit Napalm kompatibel ist (siehe Hinweise zur Ersetzung der Konfiguration File bei Napalm bei Truplshooting section).

```
mana_config():
login_choice = int(input('\n Welche Konfiguration Management Aufgabe wollen Sie erledigen ?: \n "1" Backup confguration
if login_choice == 1:
    command = 'show run '
    Netmiko_config.report(command)
```

REPLACE CONFIGRATION

Bei der Ersetzung der Konfiguration wird nach der Auswahl der neuen Konfigurationsdatei (über einen Dateiauswahldialog) ein Vergleich zwischen der vorhandenen Geräte-Konfiguration und der neuen Konfiguration durchgeführt. Änderungen in der neuen Konfiguration werden mit einem (+) Zeichen markiert, während Bestandteile, die nur in der Geräte-Konfiguration vorhanden sind und nicht in der neuen Datei, mit einem (-) Zeichen gekennzeichnet werden. Der Administrator wird gefragt, ob er die neuen Änderungen bestätigen möchte. Anschließend hat er die Wahl, die Änderungen zu bestätigen (Commit Configuration) oder die Änderungen abzulehnen. Bei einer Ablehnung wird automatisch ein Rollback zur ursprünglichen, gespeicherten Konfiguration durchgeführt.

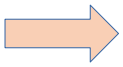
**Voraussetzungen für die neue Konfiguration(sehe bitte noch Section Troupleschouting):

1-Beginnen mit "Version" und enden mit "Ende": Die neue Konfigurationsdatei muss mit "Version" beginnen und mit "Ende" enden.

2-Keine Sonderzeichen: Die Datei darf keine Sonderzeichen enthalten. Beispielsweise wird das Symbol (^) in Banner-Definitionen vom System abgelehnt und führt zu einem Fehler (Exception Error).

Hinweis: Eine spezielle ASCII-Definition wird in diesem Programm nicht verwendet, um die Komplexität zu vermeiden. Diese Funktion könnte jedoch bei Bedarf hinzugefügt werden. Für ein Beispiel einer Konfigurationsdatei siehe die Projektdokumentation (replace_config.cfg).

*hier wird die Merge Configuration bei SW3 durchgeführt (sieht man bei Linken Seite mehr +&- über bestehende&new Configuration



```
 "5" quit
>>2

 Welche Konfiguration Management Aufgabe wollen Sie erledigen ?: ,
 "1" Backup confguration ,
 "2" Replace Configuration ,
 "3" Merge Configuration ,
 "4" Restore last Configuration(after merge or Reolace Configuration Process) ,
 "5" quit
>>2
connecting to Host192.168.52.150
applied the new configuration
die neue Konfiguration ist :
>>
+username dunya secret 5 $1$u5xe$QUooNpZfpJLW2vGW8QLx4.
-username cisco3 secret 5 $1$u5xe$QUooNpZfpJLW2vGW8QLx4.
-archive
 -path flash2:
-interface Port-channel1
 -switchport trunk encapsulation dot1q
 -switchport mode trunk
-interface Port-channel2
 -no switchport
 -ip address 192.168.20.1 255.255.255.0
-banner exec ^C
**********************************************************************
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any       *
* purposes is expressly prohibited except as otherwise authorized by     *
* Cisco in writing.                                                      *
**********************************************************************^C
-banner incoming ^C
**********************************************************************
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any       *
* purposes is expressly prohibited except as otherwise authorized by     *
* Cisco in writing.                                                      *
**********************************************************************^C
-banner login ^C
**********************************************************************
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


 connecting to Host192.168.52.150
applied the new configuration
die neue Konfiguration ist :
>>
-banner exec ^C
**************************************************************************
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any       *
* purposes is expressly prohibited except as otherwise authorized by     *
* Cisco in writing.                                                      *
**************************************************************************^C
-banner incoming ^C
**************************************************************************
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any       *
* purposes is expressly prohibited except as otherwise authorized by     *
* Cisco in writing.                                                      *
**************************************************************************^C
-banner login ^C
**************************************************************************
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any       *
* purposes is expressly prohibited except as otherwise authorized by     *
* Cisco in writing.                                                      *
**************************************************************************^C
Done,the configuration is committed
disconnecting from Host192.168.52.150...

Process finished with exit code 0
```

MERG-CONFIGRATION

Bei der Zusammenführung der Konfiguration (Merge Configuration) wird nach der Auswahl der neuen Konfigurationsdatei (über einen Dateiauswahldialog)ein Vergleich zwischen der vorhandenen Geräte-Konfiguration und der neuen Konfiguration durchgeführt. Änderungen in der neuen Konfiguration werden mit einem (+) Zeichen markiert.

Der Administrator wird gefragt, ob er sich der neuen Änderungen sicher ist. Anschließend hat er die Wahl, die Änderungen zu bestätigen (Commit Configuration) oder die Änderungen abzulehnen. Bei einer Ablehnung wird automatisch ein Rollback zur ursprünglich gespeicherten Konfiguration durchgeführt.

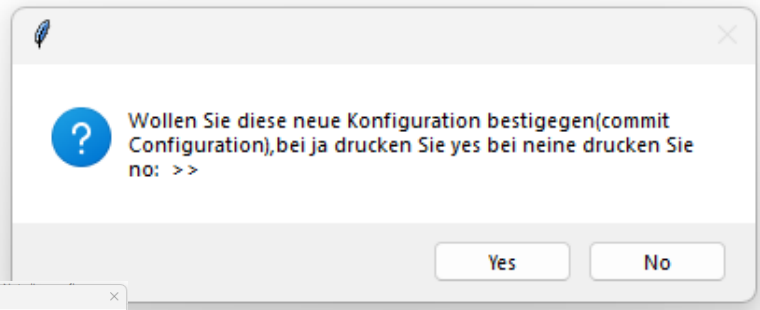
Ex: **Etherchannel L2 und L3** Erstellung bei SW3 durch Merg Configuration

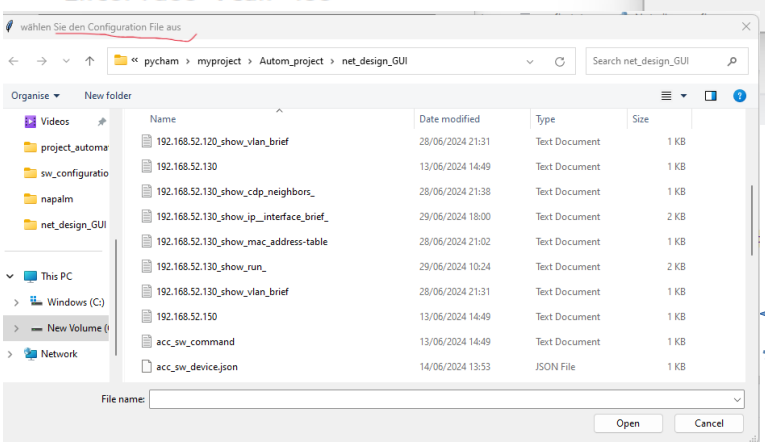
```
root@NetworkAutomation-Dunya:~/python_project# python3 napa_config.py
connecting to Host192.168.52.150
die neue Konfiguration ist :
>>
+interface range g2/0-3
+channel-group 1 mode desirable
+interface port-channel 1
+switchport trunk encapsulation dot1q
+switchport mode trunk
+no shut
+interface range g3/0-3
+channel-group 2 mode desirable
+interface port-channel 2
+no switchport
+ip address 192.168.20.1 255.255.255.0
+no shut
Wollen Sie diese neue Konfiguration bestigegen(comit Configuration),bei ja drucken Sie y bei neine drucken Sie n:
 >>y
Done
disconnecting from Host192.168.52.150
```

```
Port-channel2          192.168.20.1     YES NVRAM  down                    down
Vlan1                  192.168.52.150   YES NVRAM  up                      up
Vlan2                  10.10.10.1       YES TFTP   administratively down down
Sw3#show run | sec user
username cisco privilege 15 secret 5 $1$UUlo$2A8Whyyc7wwSI4cXkYyDT/
username cisco3 secret 5 $1$u5xe$QUooNpZfpJLW2vGW8QLx4.
username dunya secret 5 $1$lOW9$iQGRYonFIi.xZhZeHcCQB1
```

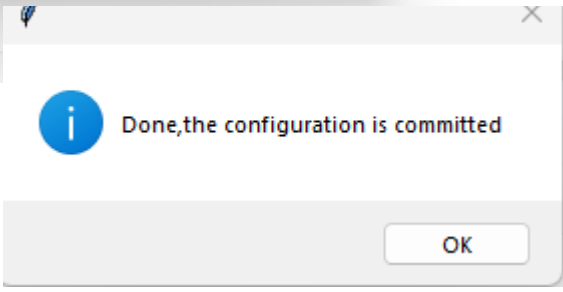
Ex2: andere Beispiel bei Add Vlans to the 2 switches SW1 & SW3

```
>>
+vlan 400
+ name Datatest
+vlan 300
+ name Voicetest
+vlan 105
+ name Test
+interface vlan 400
+ description In-band Management
disconnecting from Host192.168.52.150
connecting to Host192.168.52.120
die neue Konfiguration ist :
>>
+vlan 105
+ name Test
+interface vlan 400
```



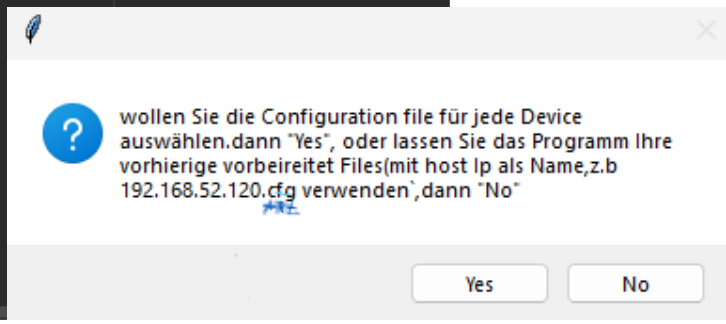


*Comit Configuration Dialog box



*open Dialog Window für die Configuration File

```python
1 usage
def merge_config():
    # ask the user if he want to give the file from openBox or select the file according to hostIp
    ask = messagebox.askquestion(message='wollen Sie die Configuration file für jede Device auswählen.dann  "Yes", oder lassen Sie das Program Ihre vorhierige vorbeir
    json_file=filedialog.askopenfilename(title='wählen Sie den json File,mit Device Liste ,aus',initialdir='.')#,filetypes=("Json File","*.json")
    with open(json_file, 'r') as f:
        user_data = json.load(f)
    for item in user_data:
        host = item['host']
        user = item['username']
        password = item['password']
        secret = item['secret']
        net_driver = item['net_driver']
        optional_args = {'secret': secret}
        print(f'\n connecting to Host{host}')
        device_ios = open_connection(host, user, password, secret, net_driver,optional_args)  # open ssh connection to each device& returen value from open_connection
        if ask =='yes':
            filename = filedialog.askopenfilename(title='wählen Sie den Configuration File aus')#filetypes='*.*'
        elif ask == 'no':
            filename=f'{host}.txt'
        device_ios.load_merge_candidate(filename)  # load the config file

        diffrent = device_ios.compare_config()  # compare the new configuration with the old one
        if len(diffrent) > 0:
            print('die neue Konfiguration ist : \n>> ')
            print(diffrent)
            while True:
                answer =messagebox.askquestion (message='Wollen Sie diese neue Konfiguration bestigegen(commit Configuration),bei ja drucken Sie yes bei neine drucken
                if answer == 'yes':
                    device_ios.commit_config()
                    #print(device_ios.has_pending_commit())
                    #device_ios.confirm_commit()
                    time.sleep(10)
                    messagebox.showinfo(message='Done,the configuration is committed')
                    break
                elif answer == 'no':
                    device_ios.discard_config()
                    break

        else:
            messagebox.showinfo('keine Konfiguration Unterschied ,no changes requiered ')
```



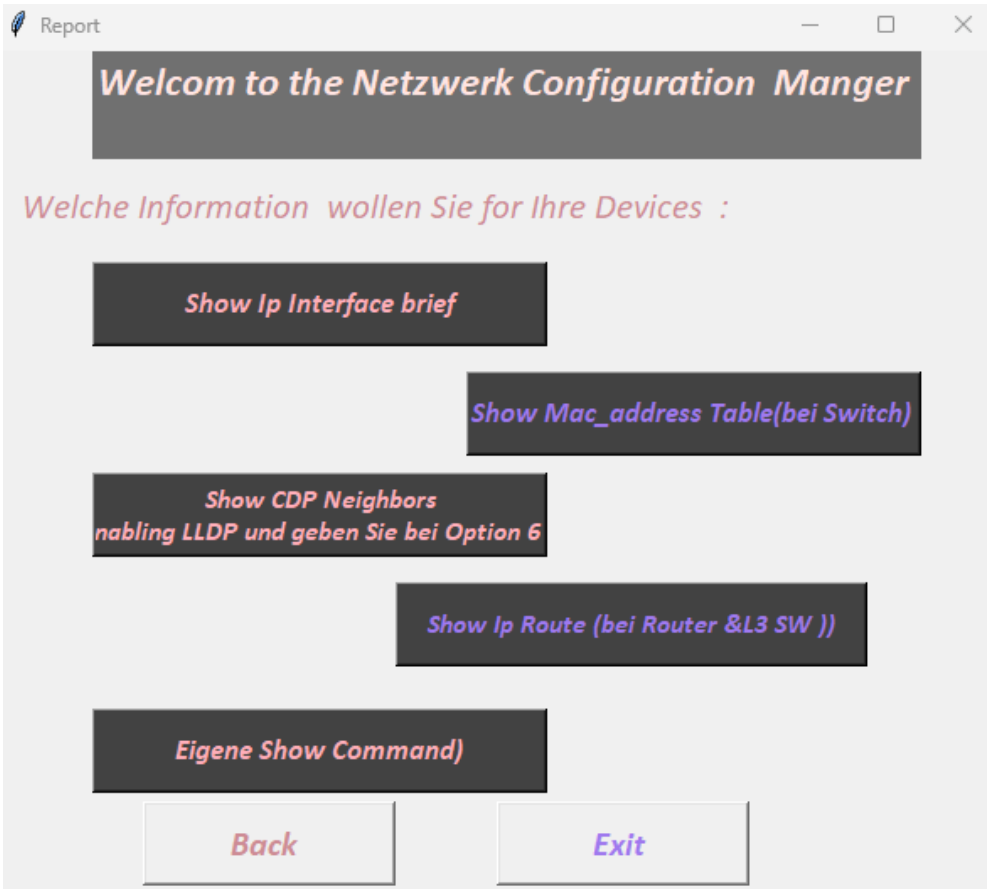
RESTORE CONFIGRATION

Hier ,bei Fehlern oder auf Wunsch wird die gespeicherte Rollback.config-Datei, die die letzte bekannte Konfiguration vor den Änderungen repräsentiert, vom System übernommen und ersetzt dabei die Running-Config auf dem Gerät.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 ****      Welcome in Network Configuration Helper       ****

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 connecting to Host192.168.52.150
Rollback the old Configuration begint
disconnecting from Host192.168.52.150

 connecting to Host192.168.52.120
Rollback the old Configuration begint
disconnecting from Host192.168.52.120

Process finished with exit code 0
```

```
272  -rw-          0  Jun 29 2024 12:20:38 +00:00  merge_config.txt
273  -rw-       2668  Jun 29 2024 13:14:54 +00:00  candidate_config.txt
274  -rw-       4703  Jun 29 2024 13:15:04 +00:00  rollback_config.txt
```

REPORTS



Ich habe Netmiko als Hauptmodul verwendet, weil es mehr Plattformen und Modelle als Napalm unterstützt und mehr Flexibilität bei der Ausgabe bietet. Mit dem Befehl send_command in Netmiko kann man jede mögliche Ausgabe abrufen, sei es L2- oder L3-Informationen, verschiedene Routing-Protokolle, ACL- und Policy-Ausgaben, VPN- und VRF-Status oder andere Virtualisierungsverbindungen. Dies hängt von der Konfiguration des Geräts ab. Für mehrere Geräte im Netzwerk wird die Ausgabe auf dem Bildschirm angezeigt und als **Ausgabedatei mit Datum (form show_command_name.Datum.txt)** gespeichert.

Report Output from Host 192.168.52.120 for show_cdp_neighbors_Command

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
D - Remote, C - CVTA, M - Two-port Mac Relay

*Hier ist *Screenshots von GNS3 Topologie: für verschiedene Reports Output .einigen sind von vordefinierte Output (show ip Interface,show Mac-address , ,show ip route für Roters ,oder eigene Show command ,wie bei Beispiel show Spanning-tree summury von ,oder show Ether-channel ,show ospf topology bei Router)

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
Sw3.cisco.com	Eth 0/2	123	R S I		Gig 0/0
R2.cisco	Eth 0/3	142	R B	Linux Uni	Eth 0/0

Report Output from Host 192.168.52.150 for show_cdp_neighbors_Command

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
R1.cisco	Gig 0/1	153	R B	Linux Uni	Eth 0/0
Sw1.cisco	Gig 0/0	169	R S I	Linux Uni	Eth 0/2

Total cdp entries displayed : 2

Report Output from Host 192.168.52.120 for show_ip__interface_brief_Com

nterface	IP-Address	OK?	Method	Status	Protocol
hernet0/0	unassigned	YES	unset	up	up
hernet0/1	unassigned	YES	unset	up	up
hernet0/2	unassigned	YES	unset	up	up
hernet0/3	unassigned	YES	unset	up	up
hernet1/0	unassigned	YES	unset	up	up
hernet1/1	unassigned	YES	unset	up	up
hernet1/2	unassigned	YES	unset	up	up
hernet1/3	unassigned	YES	unset	up	up
hernet2/0	unassigned	YES	unset	up	up
hernet2/1	unassigned	YES	unset	up	up
hernet2/2	unassigned	YES	unset	up	up
hernet2/3	unassigned	YES	unset	up	up
hernet3/0	unassigned	YES	unset	up	up
hernet3/1	unassigned	YES	unset	up	up
hernet3/2	unassigned	YES	unset	up	up
hernet3/3	unassigned	YES	unset	up	up
an1	192.168.52.120	YES	NVRAM	up	up
an400	unassigned	YES	unset	administratively down	down

Report Output from Host 192.168.52.150 for show_ip__interface_brief_Com

nterface	IP-Address	OK?	Method	Status	Protocol
gabitEthernet0/0	unassigned	YES	unset	up	up
gabitEthernet0/1	unassigned	YES	unset	up	up
gabitEthernet0/2	unassigned	YES	unset	down	down
gabitEthernet0/3	unassigned	YES	unset	down	down
gabitEthernet1/0	unassigned	YES	unset	down	down
gabitEthernet1/1	unassigned	YES	unset	down	down
gabitEthernet1/2	unassigned	YES	unset	down	down
gabitEthernet1/3	unassigned	YES	unset	down	down
gabitEthernet2/0	unassigned	YES	unset	administratively down	down
gabitEthernet2/1	unassigned	YES	unset	administratively down	down
gabitEthernet2/2	unassigned	YES	unset	administratively down	down
gabitEthernet2/3	unassigned	YES	unset	administratively down	down
gabitEthernet3/0	unassigned	YES	unset	administratively down	down
gabitEthernet3/1	unassigned	YES	unset	administratively down	down
gabitEthernet3/2	unassigned	YES	unset	administratively down	down
gabitEthernet3/3	unassigned	YES	unset	administratively down	down
ort-channel1	unassigned	YES	unset	down	down
ort-channel2	192.168.20.1	YES	NVRAM	down	down
an1	192.168.52.150	YES	NVRAM	up	up
an400	unassigned	YES	unset	administratively down	down

Report Output from Host 192.168.52.100 for show_ip_route_Command

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

S*    0.0.0.0/0 is directly connected, Ethernet0/0
      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback1
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/11] via 192.168.52.110, 00:08:19, Ethernet0/0
      192.168.52.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.52.0/24 is directly connected, Ethernet0/0
L        192.168.52.100/32 is directly connected, Ethernet0/0
```

Report Output from Host 192.168.52.110 for show_ip_route_Command

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

S*    0.0.0.0/0 is directly connected, Ethernet0/0
      1.0.0.0/32 is subnetted, 1 subnets
O        1.1.1.1 [110/11] via 192.168.52.100, 00:07:28, Ethernet0/0
      2.0.0.0/32 is subnetted, 1 subnets
C        2.2.2.2 is directly connected, Loopback1
      192.168.52.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.52.0/24 is directly connected, Ethernet0/0
L        192.168.52.110/32 is directly connected, Ethernet0/0
```

Report Output from Host 192.168.52.120 for show_mac_address-tableCommand

```
          Mac Address Table
-------------------------------------------

lan   Mac Address       Type        Ports
--    -----------       --------    -----
 1    000c.2978.799e    DYNAMIC     Et0/1
 1    0200.4c4f.4f50    DYNAMIC     Et0/1
 1    0c41.df3a.0000    DYNAMIC     Et0/2
 1    0c41.df3a.8001    DYNAMIC     Et0/2
 1    aabb.cc00.0200    DYNAMIC     Et0/3
otal Mac Addresses for this criterion: 5
```

Report Output from Host 192.168.52.150 for show_mac_address-tableCommand

```
          Mac Address Table
-------------------------------------------

lan   Mac Address       Type        Ports
--    -----------       --------    -----
 1    000c.2978.799e    DYNAMIC     Gi0/0
 1    0200.4c4f.4f50    DYNAMIC     Gi0/0
 1    aabb.cc00.0100    DYNAMIC     Gi0/1
 1    aabb.cc00.0200    DYNAMIC     Gi0/0
 1    aabb.cc00.0320    DYNAMIC     Gi0/0
otal Mac Addresses for this criterion: 5
```

Report

Report Output from Host 192.168.52.100 for show_ip_ospf_topo

OSPF Router with ID (1.1.1.1) (Process ID 1)

Base Topology (MTID 0)

Topology priority is 64
Router is not originating router-LSAs with maximum metric
Number of areas transit capable is 0
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPFs 10000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Area BACKBONE(0)
SPF algorithm last executed 00:16:02.216 ago
SPF algorithm executed 2 times
Area ranges are

Report Output from Host 192.168.52.110 for show_ip_ospf_topo

OSPF Router with ID (2.2.2.2) (Process ID 1)

Base Topology (MTID 0)

Topology priority is 64
Router is not originating router-LSAs with maximum metric
Number of areas transit capable is 0
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPFs 10000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Area BACKBONE(0)
SPF algorithm last executed 00:16:03.371 ago
SPF algorithm executed 5 times
Area ranges are

Report

Report Output from Host 192.168.52.120 for show_Spanning-tree__sumCommand

Switch is in pvst mode
Root bridge for: VLAN0002-VLAN0006, VLAN0100-VLAN0102, VLAN0105, VLAN0300
VLAN0400
Extended system ID is enabled
Portfast Default is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default is disabled
EtherChannel misconfig guard is enabled
Configured Pathcost method used is short
UplinkFast is disabled
BackboneFast is disabled

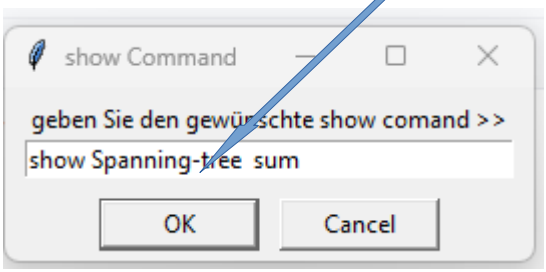
Name	Blocking	Listening	Learning	Forwarding	STP Active
VLAN0001	0	0	0	16	16
VLAN0002	0	0	0	1	1
VLAN0003	0	0	0	1	1
VLAN0004	0	0	0	1	1
VLAN0005	0	0	0	1	1
VLAN0006	0	0	0	1	1
VLAN0100	0	0	0	1	1
VLAN0101	0	0	0	1	1
VLAN0102	0	0	0	1	1
VLAN0105	0	0	0	1	1
VLAN0300	0	0	0	1	1
VLAN0400	0	0	0	1	1
12 vlans	0	0	0	27	27

Report Output from Host 192.168.52.150 for show_Spanning-tree__sumCommand

Switch is in pvst mode
Root bridge for: VLAN0001
Extended system ID is enabled
Portfast Default is disabled
Portfast Edge BPDU Guard Default is disabled
Portfast Edge BPDU Filter Default is disabled
Loopguard Default is disabled
PVST Simulation Default is enabled but inactive in pvst mode
Bridge Assurance is enabled but inactive in pvst mode
EtherChannel misconfig guard is enabled
Configured Pathcost method used is short
UplinkFast is disabled
BackboneFast is disabled

Name	Blocking	Listening	Learning	Forwarding	STP Active

*Beispiel von Eigene Show Command (show ip Ospf Topology bei Router R1&R2 ,show spanning-tree Summry bei SW1&SW3)



*Liste von den txt Files,die währen Report Output Parallel gespeichert ,mit HostIp_ShowCommand_Datum.(ich habe datum später hinzugefügt ,deswegen nicht alle File mit datum)

*Python Code

```python
def report(command: str):
    end_name = command.replace( _old: ' ', _new: '_')  # give name to the report from the show comand

    report = ''
    json_file = filedialog.askopenfilename(title='wählen Sie den json File,mit Device Liste ,aus', initialdir='.')
    with open(json_file, 'r') as f:
        device_list = json.load(f)
    for item in device_list:
        host=item['host']
        user = item['username']
        if item['username'] != ' ':
            pass
        else:
            item['username'] = simpledialog.askstring( title: 'User Name',  prompt: f'enter User name for {host}:')
            # user = input(f'enter User name for {host}:\n>>')
        if item['password'] != ' ':
            pass
        else:
            item['password'] = simpledialog.askstring( title: 'Password',  prompt: f'bitte enter das Password  for user {user}  :', show='*')
            # getpass.getpass(prompt=f'bitte enter user {user} Password :\n >>'))

        if item['secret'] != ' ':
            pass
        else:
            item['secret'] = simpledialog.askstring( title: 'Secret Password',  prompt: f'bitte enter Enable Password for Host {host} :',show
            # getpass.getpass(prompt=f'bitte enter Enable Password for Host {host} :\n >>'))
        host = item['host']
        print(f'connecting to Host {host}')
        try:
            net_connect = ConnectHandler(**item) #ssh connection
            net_connect.enable()  # entering the enable mode
            output = net_connect.send_command(command, )  #send show command
            print(output)
            date=datetime.date.today()# Today Date
            filename = f'{host}_{end_name}_{date}.txt'# Name from Host_IP_show command txt_Date
            with open(filename, 'w') as f:  # # saving the outputs to files
                f.write(output)

            print(f'connection to Host{host} will be disconnect')
            net_connect.disconnect()
            report = report + f'\n \n  Report Output from Host  {host}  for {end_name}Command \n \n ' + output
        except:
```

Passwordscheck

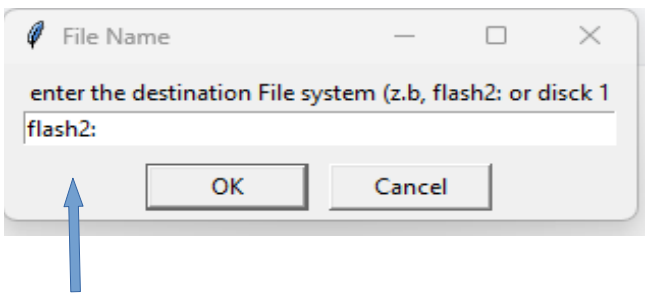
*Hier ist den Python-Skript, das eine SSH-Verbindung mit einem Netzwerkgerät herstellt, vordefinierte oder benutzerdefinierte show-Befehle ausführt und die Ausgabe zurückgibt.

➢ SCP (SECURE COPY)

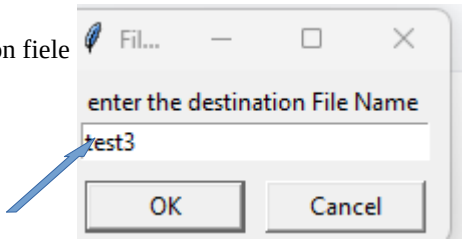
Hier wird erneut Netmiko als ConnectHandler verwendet (als SSH-Verbindungsersteller) und auch für den Dateitransfer über das Netmiko File-Transfer-Modul. Die Automatisierung des Dateitransfers vereinfacht den Firmware-Upgrade-Prozess der Geräte und ist der erste Schritt bei der Automatisierung des gesamten Upgrade-Prozesses.

Bei den Geräten muss SCP aktiviert sein (**ip scp server enable**), und es muss ausreichend Speicherplatz verfügbar sein (z.B. genug Platz im Flash-Speicher). Auch der Benutzer muss bei der Authentifizierung über **Privileg 15** (voller Administrator zugriff) verfügen.

Im Programm müssen die Quell- sowie die Zieldatei (mit optionalem Namen) angegeben werden, ebenso das Ziel-Dateisystem, je nach Gerätesystem, z.B. flash0 oder flash1 (man kann das Dateisystem und den verfügbaren Speicherplatz mit dem Befehl **dir** überprüfen).Hier wird auch Json File für die Devices benötigt ,so wird die Transfer File Prozess für gesamt Netzwerk Geräte(bei Wunsch) durchgeführt ,Falls dise Geräte bei Netmiko Bibliothek unterstützt

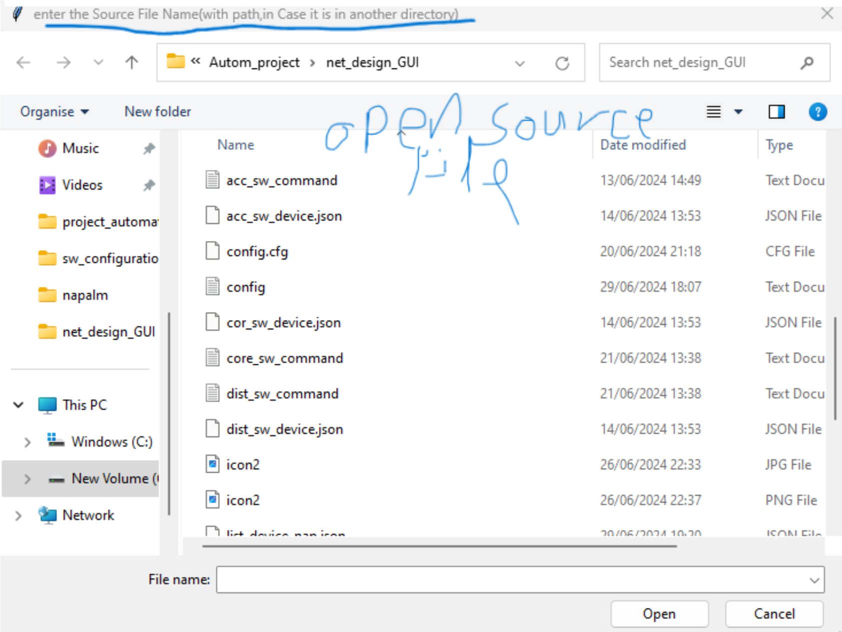


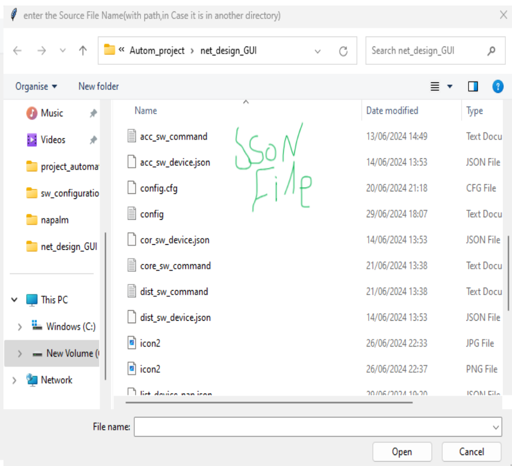
Destination fiele



***Output from Transfer process:**

```
****       Welcome in Network Configuration Helper        ****

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 connecting to Host 192.168.52.150 and transfer file
SSH connection established to 192.168.52.150:22
Interactive SSH session established
{'file_exists': True, 'file_transferred': True, 'file_verified': True}

 connection to Host192.168.52.150 will be disconnect

 connecting to Host 192.168.52.120 and transfer file
SSH connection established to 192.168.52.120:22
Interactive SSH session established

 the Host 192.168.52.120 is DOWN!!

Process finished with exit code 0
```





```
w3#dir all
irectory of flash0:/

   1  drw-           0  Jan 30 2013 00:00:00 +00:00  boot
 264  drw-           0  Oct 14 2013 00:00:00 +00:00  config
 267  -rw-   118966680  Apr 23 2019 00:00:00 +00:00  vios_l2-adventerprisek9-m
 268  -rw-      524288  Jun 20 2024 11:55:50 +00:00  nvram
 269  -rw-         325  Jun 29 2024 12:19:16 +00:00  e1000_bia.txt
 270  -rw-          35  Jun 20 2024 12:00:50 +00:00  pnp-tech-time
 271  -rw-       32400  Jun 20 2024 12:00:52 +00:00  pnp-tech-discovery-summary
 272  -rw-           0  Jun 29 2024 12:20:38 +00:00  merge_config.txt
 273  -rw-        2668  Jun 29 2024 13:14:54 +00:00  candidate_config.txt
 274  -rw-        4703  Jun 29 2024 13:15:04 +00:00  rollback_config.txt

142715904 bytes total (2018566144 bytes free)
irectory of flash2:/

   1  -rw-          89  Jun 25 2024 08:58:40 +00:00  test.txt
   2  -rw-         958  Jun 25 2024 10:45:26 +00:00  test2.txt
   3  -rw-        4626  Jun 25 2024 16:33:44 +00:00  -Jun-25-16-33-44.092-0
   4  -rw-        4626  Jun 25 2024 16:42:50 +00:00  -Jun-25-16-42-50.128-0
   5  -rw-        4626  Jun 25 2024 16:54:36 +00:00  -Jun-25-16-54-36.299-0
   6  -rw-        4703  Jun 29 2024 13:01:34 +00:00  -Jun-29-13-01-33.761-0
   7  -rw-        4703  Jun 29 2024 13:11:08 +00:00  -Jun-29-13-11-07.744-0
   8  -rw-        4703  Jun 29 2024 13:12:56 +00:00  -Jun-29-13-12-54.667-0
   9  -rw-        4703  Jun 29 2024 13:15:06 +00:00  -Jun-29-13-15-05.204-0
  10  -rw-      188233  Jun 30 2024 13:04:50 +00:00  test3
  11  -rw-      188233  Jun 30 2024 13:11:10 +00:00  test4
```

*Python Code

```
def scp_file_transfer():
    json_file = filedialog.askopenfilename(title='wählen Sie den json File,mit Device Liste ,aus', initialdir='.')
    with open(json_file, 'r') as f:
        dev_list = json.load(f)
    for item in dev_list:
        host = item['host']
        sourcefile = filedialog.askopenfilename(title='enter the Source File Name(with path in Case it is in another directory) ', initialdir='.')
        destinationfile = simpledialog.askstring( title: 'File Name', prompt: 'enter the destination File Name ')
        filesystem = simpledialog.askstring( title: 'File Name', prompt: 'enter the destination File system (z.b, flash2: or disck 1')
        try:
            print(f'\n connecting to Host {host} and transfer file ')
            net_connect = ConnectHandler(**item)  # connect to device uber ssh
            transfer = file_transfer(net_connect, source_file=sourcefile, dest_file=destinationfile,
                                     file_system=filesystem, direction='put', overwrite_file=True, )  # scp identifacation
            print(transfer)
            print(f'\n connection to Host{host} will be disconnect')
            net_connect.disconnect()

        except:
            print(f'\n the Host {host} is DOWN!!\n')

    quit()
```


➢ MÖGLICHE TROUBLESHOOTING (FEHLERBEHEBUNG)& WEITERENTWICKLUNG AUF DEM PROGRAMM

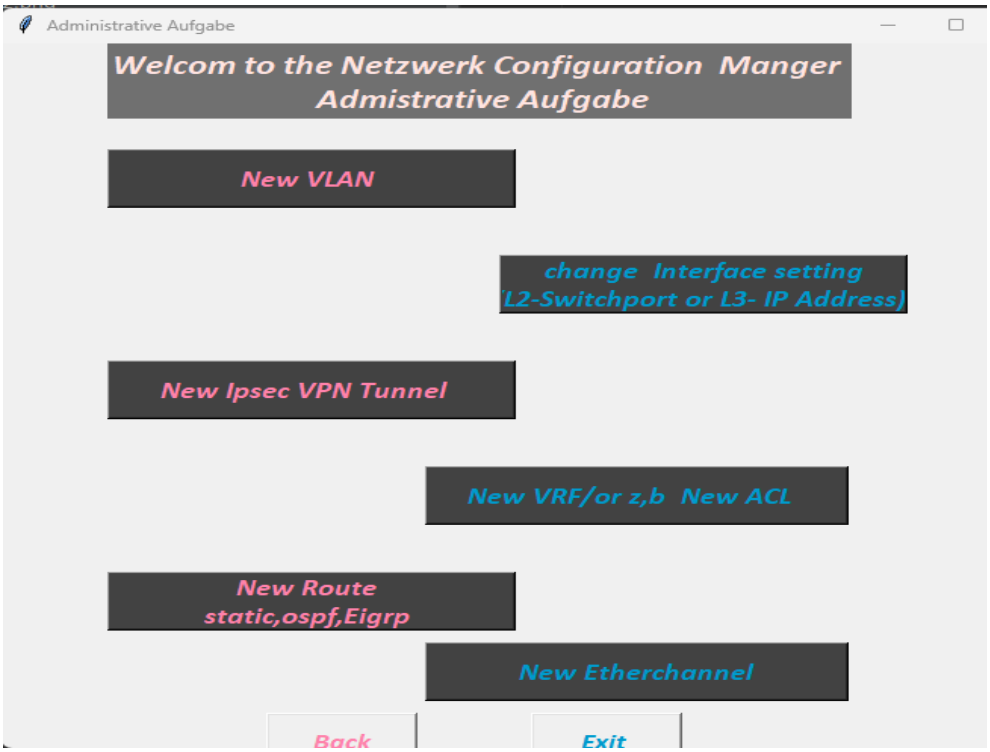
Verbesserung &Weiterentwicklung :

** Das Programm wurde bisher nur in einer virtuellen Umgebung getestet. In der realen Praxis ist es vorstellbar, dass mehrere Punkte verbessert oder geändert werden müssen, z.B. Timeout-Einstellungen. Dafür Ich bin offen für Ihre Vorschläge.

1-Ein wichtiger Punkt für den praktischen Einsatz ist jedoch das Threading (Multithreading bzw. Multiprocessing). Das bedeutet, dass anstatt auf einen einzelnen Prozess auf einem Gerät zu warten, die Prozesse parallel auf mehreren Geräten gleichzeitig ablaufen. Das macht das Programm schneller, könnte aber gleichzeitig zu einer erhöhten Komplexität beim Troubleshooting führen. Daher wurde in der Testphase auf dieses Modell verzichtet, aber das Modul könnte bei Bedarf hinzugefügt werden. :
https://realpython.com/intro-to-python-threading/

2- Es könnte eine weitere Funktion zu dem Programm hinzugefügt werden, die administrative Aufgaben erfüllt und spezifische Bedürfnisse im Netzwerk berücksichtigt. Diese Funktion könnte sich auf die Netzwerkkonfiguration und den Betrieb von Diensten in kleinen bis mittleren Netzwerkumgebungen konzentrieren. Ziel ist es, hochqualifizierte Gerätkommandos durch eine benutzerfreundliche GUI zu ersetzen.

Ein Beispiel hierfür wäre



Fehlerbehebung:

bei Netmiko (New Configuration ,Report,Backup) :

Der Einsatz von Netmiko im Programm ist einfach dargestellt, aber mögliche Fehler sollten beachtet werden:

Vor allem muss die Verbindung über SSH zuerst von dem Gerät, auf dem das Programm läuft, zu den Zielgeräten überprüft werden, um Konnektivitätsprobleme auszuschließen.
Als nächstes sollte die JSON-Eingabe überprüft werden (richtiger Gerätetreiber und Authentifizierungsinformationen).
Bei SCP muss ein Benutzer mit Privileg 15 erstellt und SCP aktiviert werden. Bei der Konfiguration könnten ebenfalls Fehler auftreten, falls die Befehle nicht zum Gerätemodell passen. Dies stoppt jedoch nicht den gesamten Prozess, wie es bei Napalm der Fall ist.

bei Napalm (section Manage Configuration) ;

-Bei Napalm, wie erwähnt, könnte IOS bei ungeeigneten Befehlen empfindlich reagieren und einen Ausnahmefehler auslösen, der das gesamte Programm stoppt.
-Ein Archiv muss auf dem Gerät definiert werden, um die Napalm-Datei zu speichern (siehe unten Hinweise zur Napalm-Programmierung).

Archive

IOSDriver requires that the *archive* functionality be enabled to perform auto-rollback on error. Make sure it's enabled and set to a local filesystem (for example 'flash:' or 'bootflash:':

```
archive
  path flash:archive
  write-memory
```

Configuration file

- IOS requires config file to begin with a *version* eg. *15.0* and *end* marker at the end of the file. Otherwise IOS will reject *configure replace* operation.
- For the diff to work properly, indentation of your candidate file has to exactly match the indentation in the running config.
- Finish blocks with *!* as with the running config, otherwise, some IOS version might not be able to generate the diff properly.

LINK TO PROJECT FILE :

- https://drive.google.com/drive/folders/1Q51SOuOYasRVxFkl3C56g2__D_HrYMH4op?usp=drive_link

**SCHLIEßLICH :

ich bedanke mich bei Ihnen für Lesen diese Dokument

* *bei Python Code habe ich keine KI Hilfe eingesetzt , aber bei Erstellung dieses Dokument habe ich ChatGPT ,als Englich Deutsch Editor angewendet

DUNYA ABDULRAZZAQ
07.2024

PYTHON CODE (OHNE GUI)

```
import Netmiko_config as Netmiko_config
import Napal_config as Napal_config
# Rahmen
text = 'Welcome in Network Configuration Helper'
space = " "
stern = "*"
def rahmen():
    lang = len(text)
    sternlang = lang + 6
    for i in range(sternlang):
        print('*', end=' ')

rahmen()
print('\n {1}{1}{1}{1}{2}{2}{2}{2}{2} {0} {2}{2}{2}{2}{2} {1}{1}{1}{1}\n'.format( *args: text, '*', ' '))
rahmen()

2 usages
def report_Choice():
    report_choice = int(input('\n Welche Information  wollen Sie for Ihre Devices  ,\n "1" Show Running-Co
    if report_choice == 1:
        command='show run '
        Netmiko_config.report(command)
    elif report_choice == 2:
        command = 'show ip interface brief '
        Netmiko_config.report(command)
    elif report_choice == 3:
        command = 'show mac address-table'
        Netmiko_config.report(command)
    elif report_choice == 4:
        command = 'show cdp neighbors '
        Netmiko_config.report(command)
    elif report_choice == 5:
        command = 'show ip route '
        Netmiko_config.report(command)
    elif report_choice == 6:
        command = input('geben Sie den gewünschte show comand ,\n>>')
        Netmiko_config.report(command)
    elif report_choice == 7:
        quit()
    else:
        print('falsche Eingabe ,versuchen Sie nochmal')
        report_Choice()
```

```
usage
def new_config(): # function for new configuration
    new_choice=int(input('\n Welche Switches or Device Wollen Sie konfigurieren ,\n "1" Core Layer Switches '
                         ' \n "2" Distribution Layer Switches  \n "3" Access Layer Switches  \n "4" Network Device acco
    if new_choice == 1:
        Netmiko_config.core_Sw()
    elif new_choice == 2:
        Netmiko_config.dist_Sw()
    elif new_choice == 3:
        Netmiko_config.acc_Sw()
    elif new_choice == 4:
        Netmiko_config.list_Sw()
    elif new_choice == 5:
        quit()
    else:
        print('falsche Eingabe ,versuchen Sie nochmal')
        new_choice()

usage
def mana_config():
    login_choice = int(input('\n Welche Konfiguration Management Aufgabe wollen Sie erledigen ?: ,'
                             '\n "1" Backup confguration ,\n "2" Replace Configuration ,\n "3" Merge Configuration ,\n
    if login_choice == 1:
        command = 'show run '
        Netmiko_config.report(command)

    elif login_choice == 2:
        Napal_config.replace_config()
    elif login_choice == 3:
       Napal_config.merge_config()
    elif login_choice == 4:
        Napal_config.restore_config()
    elif login_choice == 5:
        quit()
    else:
        print('falsche Eingabe ,v
        login_Choice()
usages (1 dynamic)
def login_Choice():
    login_choice = int(input('\n
    if login_choice == 1:
        new_config()
```

Netmik_config

```
from netmiko import ConnectHandler
import json
import time
from netmiko import file_transfer


4 usages
def execute(device, command_list):#open ssh connection and execute command
    try:
        net_connect = ConnectHandler(**device)
        net_connect.enable() # entering the enable mode
        output = net_connect.send_config_set(command_list)
        print(output)
        print(f'connection to Host{host} will be disconnect')
        net_connect.disconnect()
    except:
        print(f'\n the Host {device} is DOWN!!\n')
7 usages
def report(command:str):
    end_name =command.replace( _old: ' ', _new: '_')#give name to the report from the show comand
    with open('report_to_device.json', 'r') as f:
        device_list = json.load(f)
    for item in device_list:
        host = item['host']
        print(f'connecting t Host {host}')
        try:
            net_connect = ConnectHandler(**item)
            net_connect.enable()  # entering the enable mode
            output = net_connect.send_command(command, use_textfsm=True)
            print(output)
            filename = f'{host}_{end_name}.txt'
            with open(filename, 'w') as f:  # # saving the outputs to files
                f.write(output)
            print(f'connection to Host{host} will be disconnect')
            net_connect.disconnect()
        except :
            print(f'\n the Host {host} is DOWN!!\n')
```

```
def acc_Sw(): #for Acess Layer Switch
        acc_lines = f.read().splitlines()# read the command as list
    print(acc_lines)
    with open('acc_sw_device.json', 'r') as f:
        acc_sw_list = json.load(f)
    for item in acc_sw_list:
        host = item['host']
        print(f'connecting to Host {host}  ')
        execute(item,acc_lines)

1 usage
def list_Sw(): #for Switch from list with Management IP
     with open('list_device_net.json', 'r') as f:
        dev_list = json.load(f)
     for item in dev_list :
         host=item['host']
         print(f'connecting to Host {host}  ')
         with open(f'{host}.txt') as f:
            list_lines = f.read().splitlines()## read the command as list
            print(list_lines)
         execute(item,list_lines)

1 usage
def scp_file_transfer():
    with open('list_device_net.json', 'r') as f:
        dev_list = json.load(f)
    for item in dev_list:
        host = item['host']
```

Napalm config

```
import getpass
import json
import time
from napalm import get_network_driver
from tkinter import filedialog
from tkinter import messagebox
from tkinter import simpledialog

3 usages
def open_connection(host, user, password, secret, net_driver, optional_args):# the Connection Bau to the Device
    if user != ' ':
        pass
    else:
        user=simpledialog.askstring( title: 'User Name', prompt: f'enter User name for {host}:')
        #user = input(f'enter User name for {host}:\n>>')
    if password != ' ':
        pass
    else:
        password =simpledialog.askstring( title: 'Password', prompt: f'bitte enter user {user} Password :', show='*')
            #getpass.getpass(prompt=f'bitte enter user {user} Password :\n >>'))

    if secret != ' ':
        pass
    else:
        secret = simpledialog.askstring( title: f'bitte enter Enable Password for Host {host} :', show='*')
            #getpass.getpass(prompt=f'bitte enter Enable Password for Host {host} :\n >>'))

    driver = get_network_driver(net_driver)  # Driver used for different modell& network Venders  (z.b with meistens cisco sw und ro
    device_ios = driver(host, user, password,optional_args=optional_args)  # api or ssh(bei ios) connection to the device
    device_ios.open()
    return device_ios
1 usage
```