Automation file (netmiko pushes) this works against the IOSv-L2 GNS3 switch from the network automation appliance.

#!/usr/bin/env python

from netmiko import ConnectHandler

iosv\_l2\_s1 = {

'device\_type':'cisco\_ios',

'ip':'192.168.122.190',

'username':'dental',

'password':'D3ntal01',

}

net\_connect = ConnectHandler(\*\*iosv\_l2\_s1)

output = net\_connect.send\_command('show ip int brief')

print(output)

print 'sending command2'

output2 = net\_connect.send\_command('show version')

print(output2)

Installing Ansible (ssh automation)

apt-get update

sudo apt-get install software-properties

apt-add-repository ppa:ansible/ansible

sudo apt-get update

sudo apt-get install ansible

IPterm appliances are lightweight docker and have tools builtin also are Linux.

HP VSR1000v is available and is the HP version of the CSR 1000v & can be added to GNS3 for free , an eval is available for 60 days.

Telnet automation

root@NetworkAutomation-1:~# more python32.py

#! /usr/bin/env python3

import getpass

import telnetlib

HOST = "192.168.122.167"

user = input("enter your telnet username: ")

password = getpass.getpass()

tn = telnetlib.Telnet(HOST)

tn.read\_until(b"Username: ")

tn.write(user.encode('ascii') + b"\n")

if password:

tn.read\_until(b"password: ")

tn.write(password.encode('ascii') + b"\n")

tn.write(b"enable\n")

tn.write(b"02bladee\*\n")

tn.write(b"conf t\n")

tn.write(b"int loop 0\n")

print(tn.read\_all().decode('ascii'))

The above is run as python3 filename

The script will telnet into the device modify the bottom portion with the reuqired commands to configure the network device

Concatenation: (strings below)

y = "2"

x = "3"

str(y) + str(x) = 23

Addition: (integers below)

x = 2

y = 3

x + y = 5

The below insert into python 3 script name - the for loop where n is used , is used to denote the count and str(n) is used as a function to convert the integer from the loop to a string.

import getpass

import telnetlib

HOST = "192.168.122.72"

user = input("Enter your telnet username: ")

password = getpass.getpass()

tn = telnetlib.Telnet(HOST)

tn.read\_until(b"Username: ")

tn.write(user.encode('ascii') + b"\n")

if password:

tn.read\_until(b"Password: ")

tn.write(password.encode('ascii') + b"\n")

tn.write(b"enable\n")

tn.write(b"cisco\n")

tn.write(b"conf t\n")

for n in range (2,101):

tn.write(b"vlan " + str(n).encode('ascii') + b"\n")

tn.write(b"name Python\_VLAN\_" + str(n).encode('ascii') + b"\n")

tn.write(b"end\n")

tn.write(b"wr\n")

tn.write(b"exit\n")

print(tn.read\_all().decode('ascii'))

Methodology via Python, point at switches.txt , loop through with python to execute a command against all switches:

import getpass

import telnetlib

HOST = "localhost"

user = input("Enter your remote account: ")

password = getpass.getpass()

f = open ('myswitches')

for IP in f:

IP=IP.strip()

print ("Configuring Switch " + (IP))

HOST = IP

tn = telnetlib.Telnet(HOST)

tn.read\_until(b"Username: ")

tn.write(user.encode('ascii') + b"\n")

if password:

tn.read\_until(b"Password: ")

tn.write(password.encode('ascii') + b"\n")

tn.write(b"conf t\n")

tn.write(b"vlan 2\n")

tn.write(b"vlan 2\n")

tn.write(b"name Python\_VLAN\_2\n")

tn.write(b"exit\n")

tn.write(b"vlan 3\n")

tn.write(b"name Python\_VLAN\_3\n")

tn.write(b"exit\n")

tn.write(b"vlan 4\n")

tn.write(b"name Python\_VLAN\_4\n")

tn.write(b"exit\n")

tn.write(b"vlan 5\n")

tn.write(b"name Python\_VLAN\_5\n")

tn.write(b"exit\n")

tn.write(b"vlan 6\n")

tn.write(b"name Python\_VLAN\_6\n")

tn.write(b"vlan 7\n")

tn.write(b"name Python\_VLAN\_7\n")

tn.write(b"vlan 8\n")

tn.write(b"name Python\_VLAN\_8\n")

tn.write(b"end\n")

tn.write(b"exit\n")

print(tn.read\_all().decode('ascii'))

myswitches.txt contains

192.168.122.72

192.168.122.82

192.168.122.83

192.168.122.84

192.168.122.85

The above scrips combined:

import getpass

import telnetlib

HOST = "localhost"

user = input("Enter your remote account: ")

password = getpass.getpass()

f = open ('myswitches')

for IP in f:

IP=IP.strip()

print ("Configuring Switch " + (IP))

HOST = IP

tn = telnetlib.Telnet(HOST)

tn.read\_until(b"Username: ")

tn.write(user.encode('ascii') + b"\n")

if password:

tn.read\_until(b"Password: ")

tn.write(password.encode('ascii') + b"\n")

tn.write(b"conf t\n")

for n in range (2,31):

tn.write(b"vlan " + str(n).encode('ascii') + b"\n")

tn.write(b"name Python\_VLAN\_" + str(n).encode('ascii') + b"\n")

tn.write(b"end\n")

tn.write(b"wr\n")

tn.write(b"exit\n")

print(tn.read\_all().decode('ascii'))

backup configs python

import getpass

import telnetlib

user = input('Enter your telnet username: ')

password = getpass.getpass()

f = open('myswitches')

for IP in f:

IP=IP.strip()

print ('Get running config from Switch ' + (IP))

HOST = IP

tn = telnetlib.Telnet(HOST)

tn.read\_until(b'Username: ')

tn.write(user.encode('ascii') + b'\n')

if password:

tn.read\_until(b'Password: ')

tn.write(password.encode('ascii') + b'\n')

tn.write(b"terminal length 0\n")

tn.write(b"show run\n")

tn.write(b'exit\n')

readoutput = tn.read\_all()

saveoutput = open("switch" + HOST, "w")

saveoutput.write(readoutput.decode('ascii'))

saveoutput.write("\n")

saveoutput.close

Netmiko SSH command Sending

from netmiko import ConnectHandler

iosv\_l2 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.72',

'username': 'philip',

'password': 'cisco'

}

net\_connect = ConnectHandler(\*\*iosv\_l2)

output = net\_connect.send\_command('show ip int brief')

print (output)

config\_commands = ['int loop 0', 'ip address 1.1.1.1 255.255.255.0']

output = net\_connect.send\_config\_set(config\_commands)

print (output)

for n in range (2,21):

print ("Creating VLAN " + str(n))

config\_commands = ['vlan ' + str(n), 'name Python\_VLAN ' + str(n)]

output = net\_connect.send\_config\_set(config\_commands)

print (output)

for all devices connect and execute commands

from netmiko import ConnectHandler

iosv\_l2\_s1 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.72',

'username': 'philip',

'password': 'cisco'

}

iosv\_l2\_s2 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.82',

'username': 'philip',

'password': 'cisco'

}

iosv\_l2\_s3 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.83',

'username': 'philip',

'password': 'cisco'

}

all\_devices = [iosv\_l2\_s1, iosv\_l2\_s2, iosv\_l2\_s3]

for devices in all\_devices:

net\_connect = ConnectHandler(\*\*devices)

for n in range (2,21):

print ("Creating VLAN " + str(n))

config\_commands = ['vlan ' + str(n), 'name Python\_VLAN ' + str(n)]

output = net\_connect.send\_config\_set(config\_commands)

print (output)

Cisco FIle show run deploy with netmiko

from netmiko import ConnectHandler

iosv\_l2\_s4 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.84',

'username': 'philip',

'password': 'cisco',

}

iosv\_l2\_s5 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.85',

'username': 'philip',

'password': 'cisco',

}

iosv\_l2\_s6 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.86',

'username': 'philip',

'password': 'cisco',

}

with open('iosv\_l2\_cisco\_design') as f:

lines = f.read().splitlines()

print (lines)

all\_devices = [iosv\_l2\_s4, iosv\_l2\_s5, iosv\_l2\_s6]

for devices in all\_devices:

net\_connect = ConnectHandler(\*\*devices)

output = net\_connect.send\_config\_set(lines)

print (output)

config all core

from netmiko import ConnectHandler

iosv\_l2\_s2 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.82',

'username': 'philip',

'password': 'cisco',

}

iosv\_l2\_s3 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.83',

'username': 'philip',

'password': 'cisco',

}

iosv\_l2\_s4 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.84',

'username': 'philip',

'password': 'cisco',

}

iosv\_l2\_s5 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.85',

'username': 'philip',

'password': 'cisco',

}

iosv\_l2\_s6 = {

'device\_type': 'cisco\_ios',

'ip': '192.168.122.86',

'username': 'philip',

'password': 'cisco',

}

with open('iosv\_l2\_cisco\_design') as f:

lines = f.read().splitlines()

print (lines)

all\_devices = [iosv\_l2\_s4, iosv\_l2\_s5, iosv\_l2\_s6]

for devices in all\_devices:

net\_connect = ConnectHandler(\*\*devices)

output = net\_connect.send\_config\_set(lines)

print (output)

with open('iosv\_l2\_core') as f:

lines = f.read().splitlines()

print (lines)

all\_devices = [iosv\_l2\_s6, iosv\_l2\_s5, iosv\_l2\_s4, iosv\_l2\_s3, iosv\_l2\_s2]

for devices in all\_devices:

net\_connect = ConnectHandler(\*\*devices)

output = net\_connect.send\_config\_set(lines)

print (output)