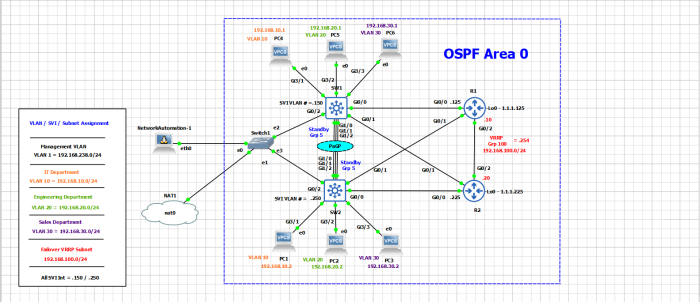
[THE CCNP GRIND!](https://loopedback.com/)

PYTHON3 NETWORK PROGRAMMING – CONFIGURING A LARGER, MORE COMPLEX TOPOLOGY USING PYTHON 3! ALL CODE / SCRIPTS USED, DEVICES EXPLAINED, AND SOME TROUBLESHOOTING! (PART 1/2)



**Its not too deep into complex configurations, but turning it up a notch**

I am still do some no-no’s like setting the “enable secret” password in plain text in the code templates, which will be something I address down the road, I’ve decided to add a few devices and a few old friends from my CCNP R/S days:

* HSRP and PaGP Port-Channel between SW1 and SW2
* VRRP configured between R1 and R2
* Enabling “ip routing” on SW1 and SW2 for OSPF and Inter-Vlan Routing
* 3 extra VLANs and VPCs to test Inter-Vlan Routing
* Added 2 hosts in each “Department” subnet to verify network segmentation
* Adding OSPF Area 0 with default network route to advertise all networks
* OSPD RID configured to make R1 DR and R2 Backup DR in OSPF Area 0
* Learned how to the GNS3 Topology look a bit more fancy!

An Important Heads up before diving into info below for readers:

I did have an outdoors day today unexpectedly (and it was niiice), however I did make unseen progress in finding misconfigs, so below is what I believe will be the updated working automation configs but I do not actually execute them in this article.

Tomorrow I will make a Part 2 where I run and troubleshoot the scripts, however this does have some good information on troubleshooting issues I actually created with GNS3, and also some good notes on configuring devices and why I did so!

So worth reading, but Part 2 will contain the actual execution of Python scripts.

**\*\*\*Also I will link to the GNS3 lab file shared via google drive in Part 2!\*\*\***

That all being said, I will post this up, and hit the real labbing next post!

**With that I will get right into it, and post the code I’ve written for each device!**

Below will be the the line of code copied directly from Visual Studio Code:

**AutoSW1**

import getpass

import telnetlib

HOST = “192.168.238.150”

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

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”loopedback\n”)

tn.write(b”conf t\n”)

tn.write(b”ip routing\n”)

tn.write(b”int lo0\n”)

tn.write(b”ip add 1.1.1.150 255.255.255.255\n”)

tn.write(b”int vlan 10\n”)

tn.write(b”description IT VLAN\n”)

tn.write(b”ip add 192.168.10.150 255.255.255.0\n”)

tn.write(b”no shut\n”)

tn.write(b”int gi3/1\n”)

tn.write(b”switchport mode access\n”)

tn.write(b”switchport access vlan 10\n”)

tn.write(b”no shut\n”)

tn.write(b”int vlan 20\n”)

tn.write(b”description Engineering VLAN\n”)

tn.write(b”ip add 192.168.20.150 255.255.255.0\n”)

tn.write(b”no shut\n”)

tn.write(b”int gi3/2\n”)

tn.write(b”switchport mode access\n”)

tn.write(b”switchport access vlan 20\n”)

tn.write(b”no shut\n”)

tn.write(b”int vlan 30\n”)

tn.write(b”description Sales VLAN\n”)

tn.write(b”ip add 192.168.30.150 255.255.255.0\n”)

tn.write(b”no shut\n”)

tn.write(b”int gi3/3\n”)

tn.write(b”switchport mode access\n”)

tn.write(b”switchport access vlan 30\n”)

tn.write(b”no shut\n”)

tn.write(b”int ra gi1/0 – 2\n”)

tn.write(b”switchport trunk encap dot\n”)

tn.write(b”switchport mode trunk\n”)

tn.write(b”channel-group 5 mode desirable\n”)

tn.write(b”int vlan 1\n”)

tn.write(b”standby 5 ip 192.168.238.5\n”)

tn.write(b”router ospf 1\n”)

tn.write(b”network 0.0.0.0 0.0.0.0 area 0\n”)

tn.write(b”router-id 1.1.1.1\n”)

tn.write(b”end\n”)

tn.write(b”wr\n”)

tn.write(b”exit\n”)

print(tn.read\_all().decode(‘ascii’))

**AutoSW2**

import getpass

import telnetlib

HOST = “192.168.238.250”

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

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”loopedback\n”)

tn.write(b”conf t\n”)

tn.write(b”ip routing\n”)

tn.write(b”int lo0\n”)

tn.write(b”ip add 1.1.1.250 255.255.255.255\n”)

tn.write(b”int vlan 10\n”)

tn.write(b”description IT VLAN\n”)

tn.write(b”ip add 192.168.10.250 255.255.255.0\n”)

tn.write(b”no shut\n”)

tn.write(b”int gi3/1\n”)

tn.write(b”switchport mode access\n”)

tn.write(b”switchport access vlan 10\n”)

tn.write(b”no shut\n”)

tn.write(b”int vlan 20\n”)

tn.write(b”description Engineering VLAN\n”)

tn.write(b”ip add 192.168.20.250 255.255.255.0\n”)

tn.write(b”no shut\n”)

tn.write(b”int gi3/2\n”)

tn.write(b”switchport mode access\n”)

tn.write(b”switchport access vlan 20\n”)

tn.write(b”no shut\n”)

tn.write(b”int vlan 30\n”)

tn.write(b”description Sales VLAN\n”)

tn.write(b”ip add 192.168.30.250 255.255.255.0\n”)

tn.write(b”no shut\n”)

tn.write(b”int gi3/3\n”)

tn.write(b”switchport mode access\n”)

tn.write(b”switchport access vlan 30\n”)

tn.write(b”no shut\n”)

tn.write(b”int ra gi1/0 – 2\n”)

tn.write(b”switchport trunk encap dot\n”)

tn.write(b”switchport mode trunk\n”)

tn.write(b”channel-group 5 mode desirable\n”)

tn.write(b”int vlan 1\n”)

tn.write(b”standby 5 ip 192.168.238.5\n”)

tn.write(b”router ospf 1\n”)

tn.write(b”network 0.0.0.0 0.0.0.0 area 0\n”)

tn.write(b”router-id 2.2.2.2\n”)

tn.write(b”end\n”)

tn.write(b”wr\n”)

tn.write(b”exit\n”)

print(tn.read\_all().decode(‘ascii’))

**AutoR1**

import getpass

import telnetlib

HOST = “192.168.238.125”

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

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”loopedback\n”)

tn.write(b”conf t\n”)

tn.write(b”int lo0\n”)

tn.write(b”ip add 1.1.1.125 255.255.255.255\n”)

tn.write(b”int gi0/2\n”)

tn.write(b”ip add 192.168.100.10 255.255.255.0\n”)

tn.write(b”vrrp 100 ip 192.168.100.254\n”)

tn.write(b”no shut\n”)

tn.write(b”router ospf 1\n”)

tn.write(b”network 0.0.0.0 0.0.0.0 area 0\n”)

tn.write(b”router-id 4.4.4.4\n”)

tn.write(b”end\n”)

tn.write(b”wr\n”)

tn.write(b”exit\n”)

print(tn.read\_all().decode(‘ascii’))

**AutoR2**

import getpass

import telnetlib

HOST = “192.168.238.225”

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

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”loopedback\n”)

tn.write(b”conf t\n”)

tn.write(b”int lo0\n”)

tn.write(b”ip add 1.1.1.225 255.255.255.255\n”)

tn.write(b”int gi0/2\n”)

tn.write(b”ip add 192.168.100.20 255.255.255.0\n”)

tn.write(b”vrrp 100 ip 192.168.100.254\n”)

tn.write(b”no shut\n”)

tn.write(b”router ospf 1\n”)

tn.write(b”network 0.0.0.0 0.0.0.0 area 0\n”)  
tn.write(b”router-id 3.3.3.3\n”)

tn.write(b”end\n”)

tn.write(b”wr\n”)

tn.write(b”exit\n”)

print(tn.read\_all().decode(‘ascii’))

**^^^Feel free to steal these and use them however you want!**

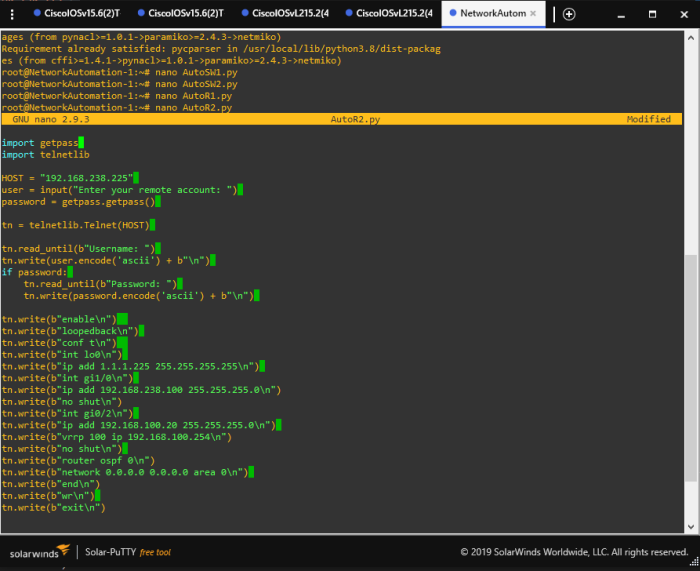
**If you’ve never wrote entire templates in notepad, you haven’t lived!**

I actually caught quite a few things I missed such as RID settings to make the Routers the DR / Backup DR for OSPF Area 0, SW2 has SW1’s SVI Gateway IP’s, and a few other small things.

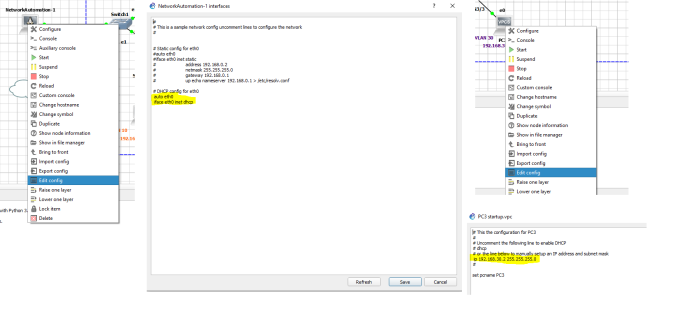
However the above code should now be able to be copied into Visual Studio Code for review, into a GNS3 Automation host device using “nano filename.py” to get a blank open window to then right click to paste it in (as SolarWinds Telnet Client that comes with GNS3 that I am using pastes just like Putty with a right-click), then ctrl+o / hit enter button (to write the file to directory) / ctrl+x and hit enter to then exit back to prompt.

These are then executed using “python3 filename.py” as shown below and probably other posts, and if you do not use python3 instead of just python in the command it will run Python 2 and may not understand some variables – And I am working with Python 3!

**This looks much easier on the eyes in Visual Studio Code or even Linux!**



**Also to quickly cover how I setup the NetworkAutomation / Virtal PC Hosts in GNS3!**



I apologize for the kind of mess of screen snips I put together, but above what is shown is me right clicking on the “NetworkAutomation-1” and going to “Edit Config” which is text, and for this I simply deleted the # in front of both DHCP statements which makes them active.

So right click Host -> delete # in front of DHCP statements -> Save -> Reboot host to pull IP.

The reason I did this is because of how the Topology is setup with a Layer 2 Switch connecting the NetAuto Host to the NAT Cloud (both created via New Template in Hosts in GNS3) allowing it to pull a DHCP IP from the NAT and allowing it to contact the local host and also get out to the internet.

Also I show the same process was done on here PC3 only I removed the # in front of ip address, and entered the VLAN Subnet / Host Address I wanted to provide it, in this case its PC3 in VLAN 30 with the IP 192.168.30.2.

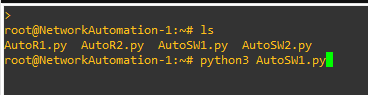
**The reason for this is I wanted to “apt-get update” my Linux host to update it OS and Python versions, and so I can SSH to the Network Devices in the same DHCP Subnet of 192.168.238.0/24 – Except of course my mgmt IP’s on Routers / Switches are statically configured in this Topology whereas I let the NetAuto Host use DHCP.**

I also had to configure the GNS3 VM’s Network segment into “Bridged Mode” to my local network, the NAT Cloud does not need special configuration beyond being on the Diagram and connected / reachable to any devices that need to reach the LAN / Internet.

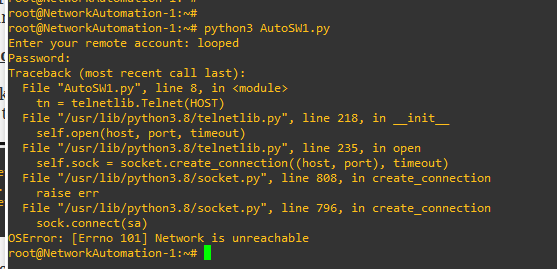
**Edit – ^^^This is wrong, see below for correct GNS3 OVA Network Settings!!!**

**That covered, I will debug Telnet on my network devices, and run some scripts!**

A quick look at how these items have been configured in the directory and the command to run them one at a time:



So here we see the 4 item in the directory via “ls” on the NetAuto-1 Host, and to execute them I will run “python3 (filename).py” one at a time:



And we have our first train wreck right off the bat!**Network Unreachable???**

In reviewing the switch I actually did not see the Telnet Connection even attempted for some reason during on the SW1 CLI, and I found SW1 and R1 still have the initial configs I automated out to them so I am doing:

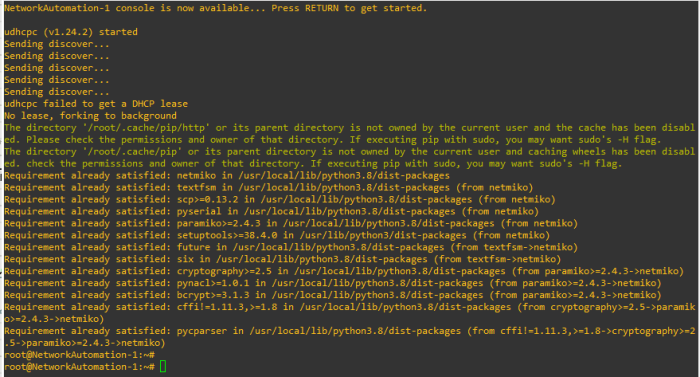
**delete vlan.dat**

**wr er**

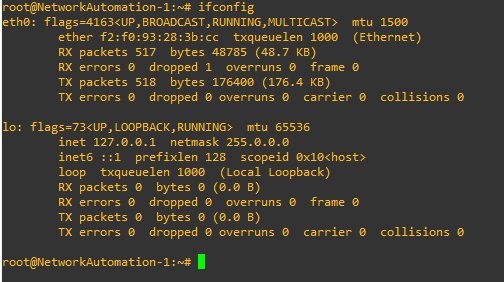
**reload**

**(Select “No” to saving changes in config)**

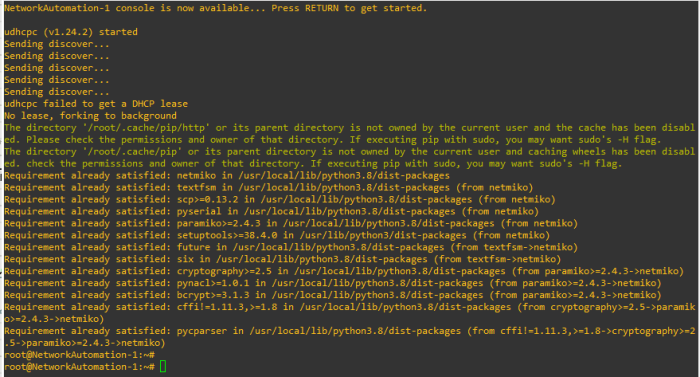
After a reload and actually completely rebooting GNS3 for a clean slate:



Now I did configure the MGMT IP on VLAN 1, so let me take a look at the NetworkAuto-1 Host device to see what if something has changed here:



Ummm…. What? Let me give this a reboot here quick!



That explains it, though I am not sure why the DHCP Server (Nat Cloud) is now failing to hand out DHCP to this network, let me look at the setting on the VM here and see if I can find the issue on the GNS3 VM.

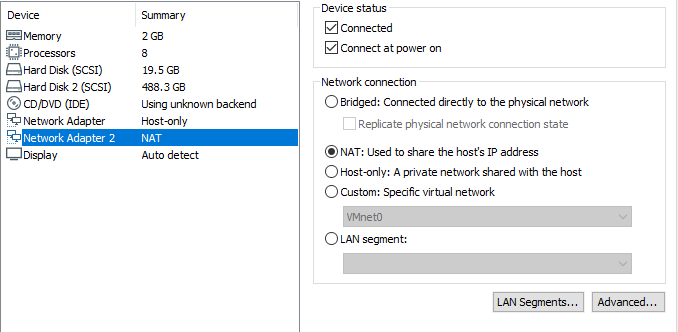
**So I found that I actually broke my network messing with the GNS3 VM Settings!**

Boy do I feel stupid

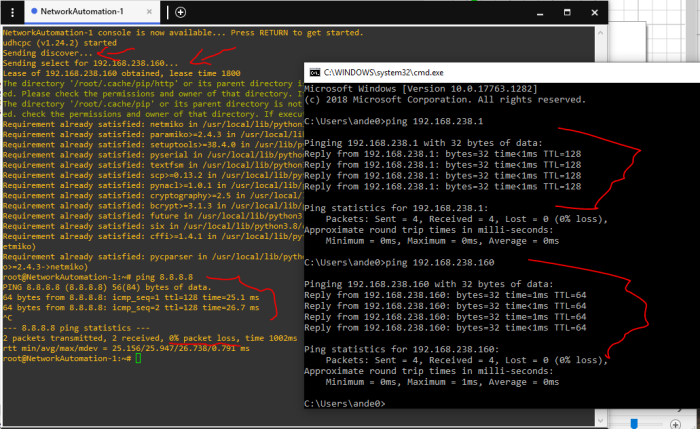
I initially had set Net Adapter #1 in the GNS3 VM Network Settings, but then for whatever reason this morning changed both Net Adapter #1 and #2 to Bridged Mode, when how the original changing of “Bridged Mode” was only working because it was falling back to Net Adapter #2 with “NAT Mode” configured when I thought “Bridged” mode just seemed to make more sense in my mind.

Though now it makes 100% sense why my home Meraki Network was not seeing any of the clients in the 192.168.238.0/24 subnet, doh!

After changing it back to its original settings of Host-Only and NAT settings for Network Adapters #1 and #2 in the GNS3 VM and rebooting my laptop, it now detects the GNS3 VM as the Server again which runs basically everything, and in checking it NAT is selected:



And then of course DHCP is working, I can ping Internet from GNS3, and my laptop can ping both the GNS3 DHCP Gateway IP and NetAuto-1 Host IP’s shown here:



So that made me feel pretty stupid initially, but you don’t learn these things until you break them, and put the work in to fix them!

**Now that I am done trying to outsmart the engineers that created the GNS3 VM OVA**

I am going to call it a night!

I have honestly spent the day enjoying the weather and just getting outside for a much needed nature day, and I was hoping to spend this time troubleshooting automation issues rather than me breaking the GNS3 VM.

However I do have plans in the early afternoon tomorrow, and I am the type of person that will not be able to sleep until I troubleshoot an issue to resolution, if my work would allow me never ending overtime I would probably work 23 hours a day (please don’t tell any of my several bosses that)!

However I actually did find some bugs that I’ve fixed in my scripts during the day which are updated in the above configs (IP Updates, added OSPF RID, etc) so I got some things corrected that were just tired typos, I hope to finish this up tomorrow!

**Happy Fathers Day to all Fathers out there, its a Holiday tomorrow (Sunday) here in the US, but to my international Fathers out there have a great Sunday!**

With that I am clocking out, and will follow through with a conclusion tomorrow!

SHARE THIS: