**Unit Nine Lab**

Directions: Complete each of the scripts below and submit them by the date specified in the assignment sheet and Blackboard. Submit **copies of your code** and **screenshots of the code running with each task**. Also, be sure to use the document, **Script Requirements as a guide** to writing good code. **Full credit will not be earned if you do not meet these script requirements.** **40 points**

**Note: Since we are running APIs on the NXOS switches in this lab, be sure to run the feature nxapi command from global configuration on each switch before attempting to send and API request.**

**NX-API Data Model**

In addition to using the NX-API to generate JSON RPC code for CLI applications, we can use the NXAPI DME Model to change or read settings and data from our switch. Submit any script files created in this lab to Blackboard.

1. Using the NXAPI to build DME model payloads and URLs for the following.
   1. Write a function for a NXOS switch that accepts a deviceIP address, a VLAN number and a VLAN name. Have the function create the VLAN.
   2. Write a function that accepts a deviceIP address, an SVI interface name, and an IP address and mask, and have it create the SVI, assign the IP, and have enabled (no shutdown)
   3. Write a function that accepts a deviceIP, interface, HSRP group, and HSRP address, and have it create the HSRP group and HSRP address on the specified device and interface (see other interfaces for NXOS configuration syntax).
   4. Write a function that accepts a deviceIP, interface, OSPF process-ID, and OSPF area, and have it add the specified interface to the process and area (see other interfaces for NXOS configuration syntax).
2. Once the above are working with the DME model, write a script that iterates through a dictionary of your NXOS switches, and pass the following information to your switches:
   1. Create a Vlan 110, named testNXOS on both switches.
   2. Create the SVI for the VLAN on each device, giving dist-SW01 the ip address of 172.16.110.2/24 and dist-SW02 the address of 172.16.110.3/24. Note that these are the second and third addresses of the range and can be automated (incremented with a variable for the 4th octet in the IP address before sending the ip address to the function created in step 1).
   3. Set up HSRP on both devices with group 10 and an HSRP address of 172.16.110.1
   4. Add each interface to OSPF process 1 and area 0.0.0.0 (area 0)
3. This project is not as large as it seems, and once you have your functions working for one device, step 2 will be easy. Also, the NXAPI will be helpful in generating the payloads for the calls.

I have provided screenshots of my postman application, I used vlan 133 for most of my testing: A screenshot of a computer

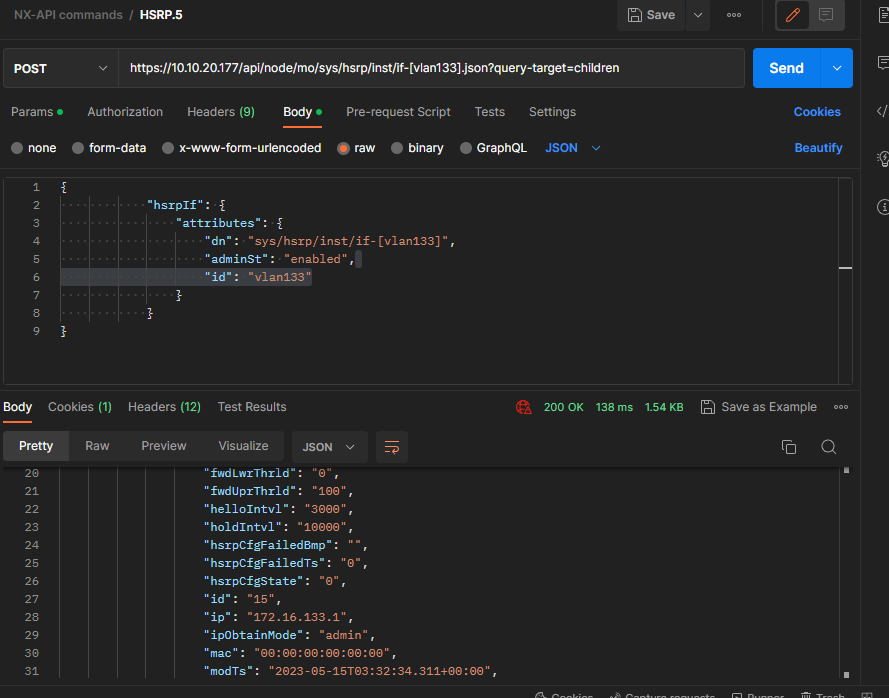
Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence



A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence