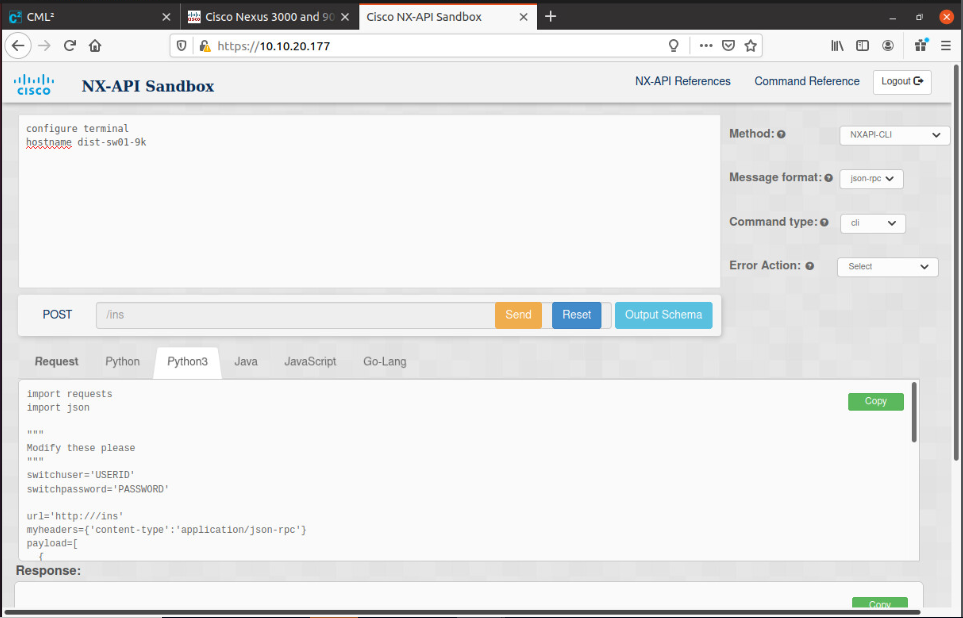
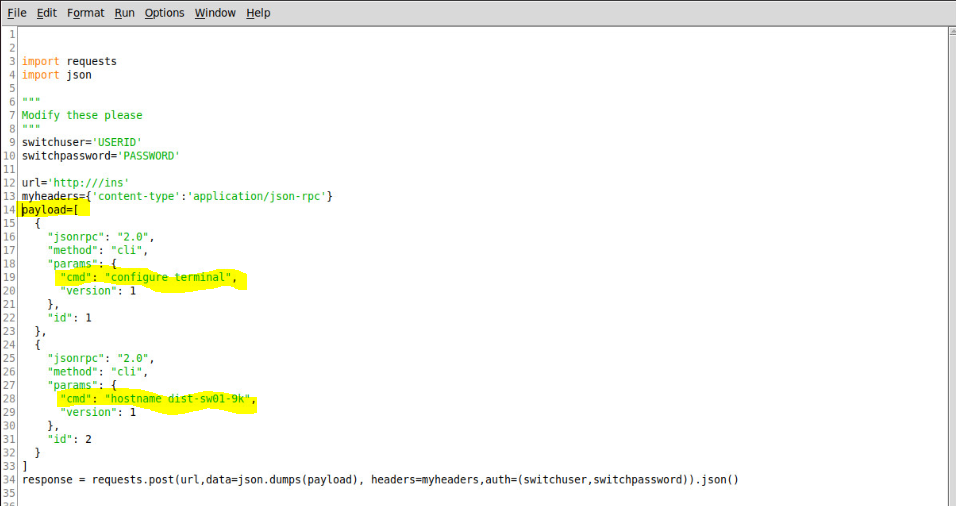
**NX-API SANDBOX 2**

Instructions: In this lab you will continue to explore the NX-API Sandbox to facilitate writing API requests to NXOS Switches. Complete the script below and submit it before the beginning of class on the date specified in the assignment sheet. 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. **10 points**

1. Make a Cisco Modeling Labs (CML) reservation
2. Once in, go to the console for the dist-sw01 switch
3. In config mode, type *feature nxapi* to enable the API interface on the switch
4. Access the NXOS Sandbox for dist-sw01.
5. In the command window, enter configure terminal and hostname dist-swo1-9k as shown below.



1. Select the Python 3 tab and then copy the code to a new program box.
2. Note that the payload variable is now a list of dictionaries, each with its own command to pass to the CLI as shown below:



1. Make the necessary changes to make your code work (the url, switchuser and switchpassword, and the verify = False parameter in the request line).
2. Run the code and go to the console of the switch to see if the host name was changed.
3. If it works, you are ready to create your custom program.
4. **When you run the script, ask the user for a new hostname**
   1. **Check that the hostname is valid (i.e., no spaces, begins with an alpha character, and does not contain special characters). It is best to pass the hostname to a function to determine if it is valid. You can have your program end with a simple error message if the user tries to enter an invalid hostname**
   2. **If the user types a valid host name, send the hostname to a function that changes the hostname on your dist-sw01 device (e.g., changeHostName(newHostName)**
   3. **Note that you will have to modify the code slightly in line 28 above for this to work. Line 28 above is a static command passed as a string. You will need to use string concatenation so that line 28 (or its equivalent in your script) looks something like:**
      1. ***‘cmd’ : ‘hostname ‘ + newHostName,***
      2. **The above line concatenates your string to so that the value stored and passed to the switch will be similar to line 28 above, except with your new name. Note that newHostName needs to be defined by you.**
      3. **You can verify that it is working by going to the console on the switch.**
5. **Write a script that asks a user for an IP address, and adds a value of two to the third octet. Then have the script print out the new IP address. For example, if you ask the user for an IP address and if they entered 10.20.20.1, your script would return 10.20.22.1. Note that the third octet incremented by 2 (20 + 2 = 22). Make sure that the incrementing code happens in a function. Also, when the user enters the IP address, make sure your code ensures a valid IP address was entered in a separate function (you have already created this code earlier in the semester).**

**Hint:** **Split the ip address string into its 4 components**

* 1. **Numerically add 2 to the third octet (You will need to cast the octet as a an integer before you can add anything to it)**
  2. **Once the third octet value is changed in the list created by the split() method and modified, you can iterate through the list and reassign the IP address to a variable with concatenation.**

**Example snippet if addressList is [“172”,”16”,”101”,”7”] and you want to have modifiedIP be “172.16.101.7”**

**modifiedIP = “ “**

**for octetVal in addressList :**

**modifiedIP = modifedIP + octetVal + “.”**

**# Don’t forget to rstrip() the last “.” after the list is iterated**

**modifiedIP = modifiedIP.rstrip(“.”)**