Purpose

The purpose of this exercise is to control your instance using a Python program from the Cloud Shell

Preparation

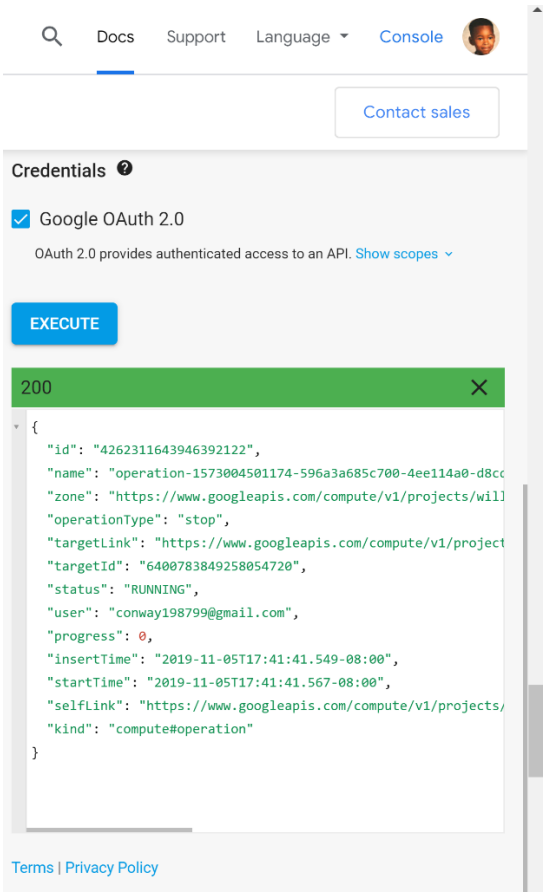
Read through this entire lab so you have a sense of what is expected of you before you start.

Read the page Methods: instances.stop (Links to an external site.), which describes the process for stopping a running instance using an API. You do not need to worry about installing any packages or authorization keys as long you run your code in the Cloud Shell.

Assignment

To document the steps below, create a Word document that contains screenshots and brief explanations of what you did. This serves as a notebook for you and a summary for the instructor to grade.

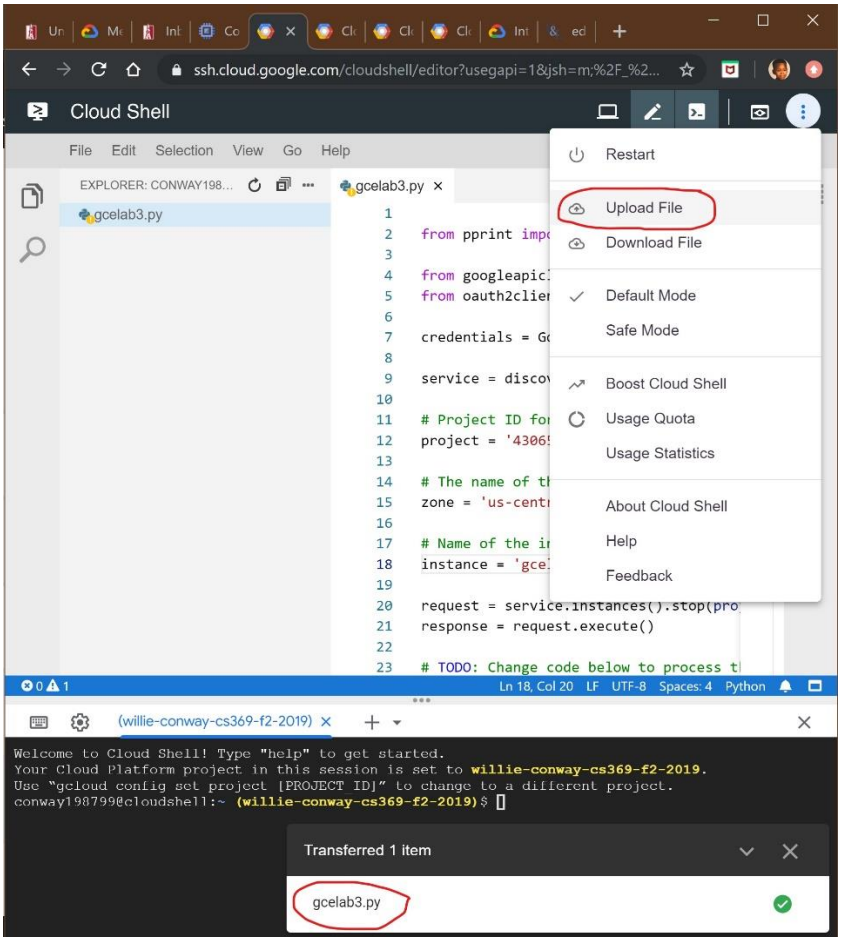
1. On the page Methods: instances.stop (Links to an external site.), use the API Explorer in section "Try this API" to test stopping your instance using an API. The three fields you need to set are project (**id** not name), zone, and instance name. Run this tool until you get the parameters correct, then take a screenshot.



(This is an output of my response dictionary after utilizing the API explorer in the section "Try this

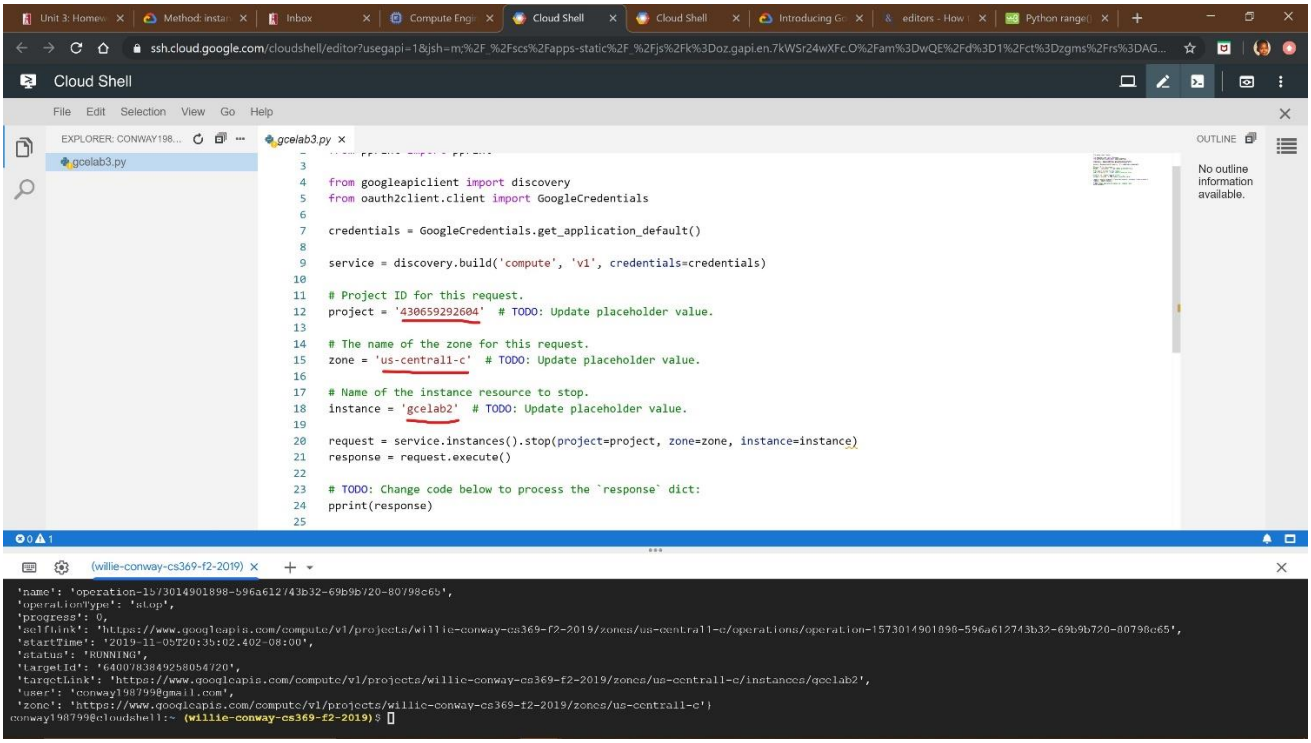
API," to test stopping my instance. To do this I entered the three fields of my project ID number, zone, and instance.)

2. Copy the python code from the section "Examples" and put it in a file on the Cloud Shell. You can either save the file on your local computer and upload the file, or you can open the Cloud Shell code editor.



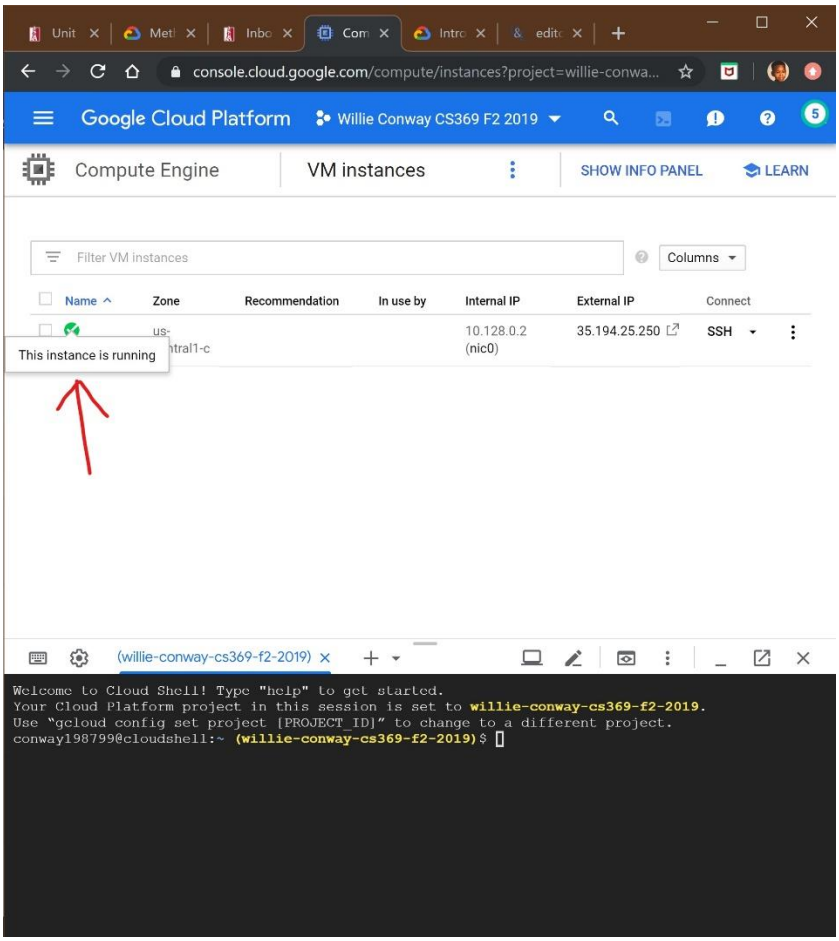
(Uploading my gcelab3.py file to the Cloud Shell Editor after writing the file in my SSH through nano editor.)

3. Edit your python program to change the sections marked TODO, to use the parameters you found in step 1.

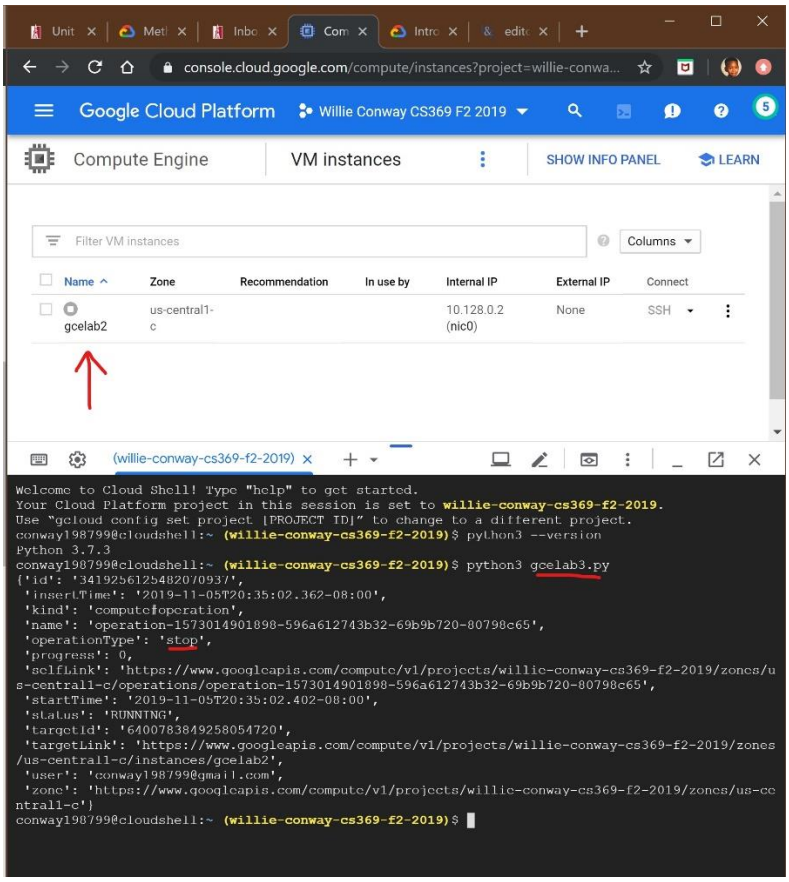


(Editing the default parameters with information from my gcelab2 instance. Entering the following field of my project ID number (430659292604), zone (us-central1-c), and instance (gcelab2). Then saving my gcelab3.py file.)

4. Run your program from the Cloud Shell command line using python3 filename.py
   Make sure your program stops a running instance. You can check this by looking at the Cloud Console.



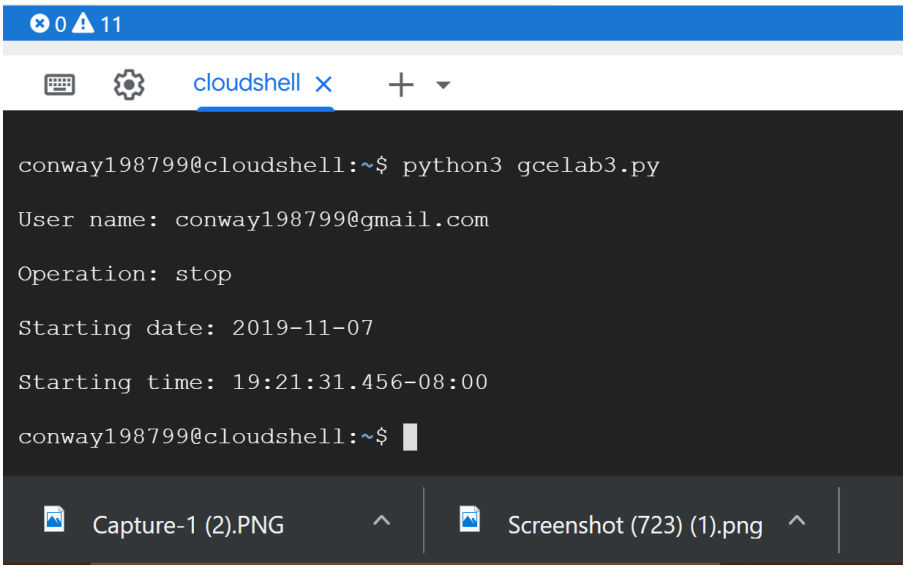
(I started my instance again so that I can test the code one more time, to make sure my fields were entered in correct, for my program to function.)



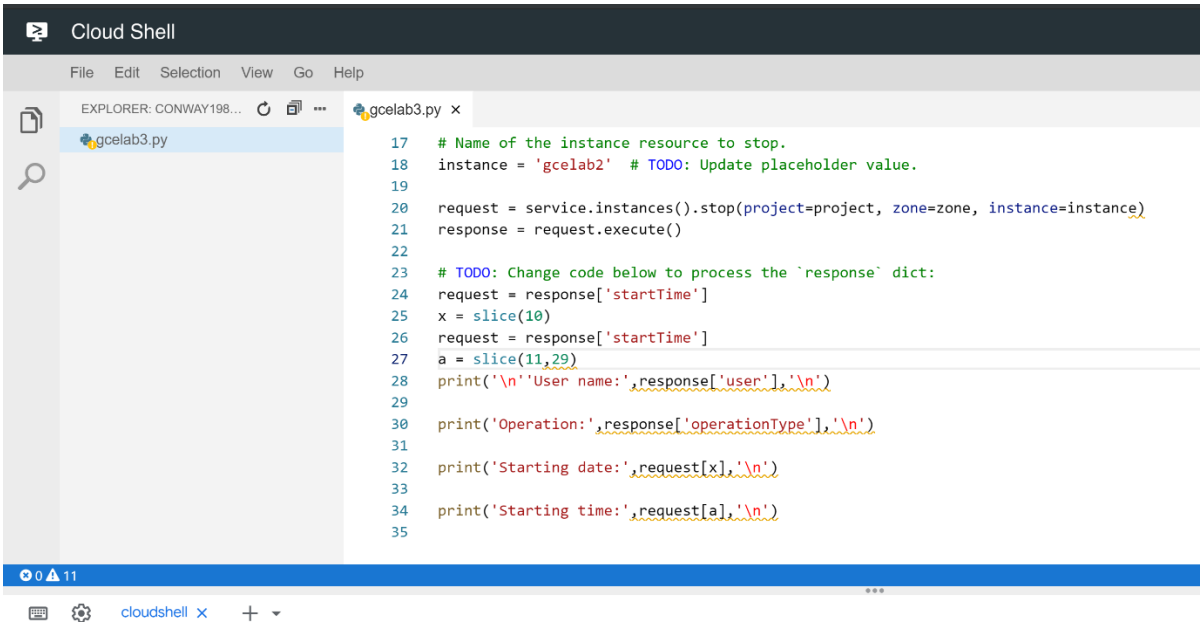
(*This is an output to show that I entered all of the following items correctly to get the same output from the website, of the complete response dictionary. It also showed that my instance gcelab2 has completely stopped. To activate this command, I entered python3 gcelab3.py. This tells the Cloud Shell that I'm using python version 3 and that I want to open the file gcelab3.py.)*

5. Modify your program so that instead of printing the entire dictionary, response, it only prints the following, each on a separate line: user, operation type, start date, start time

Since response is a dictionary, you access the elements like response['user']. Take a screenshot of the output of your program.



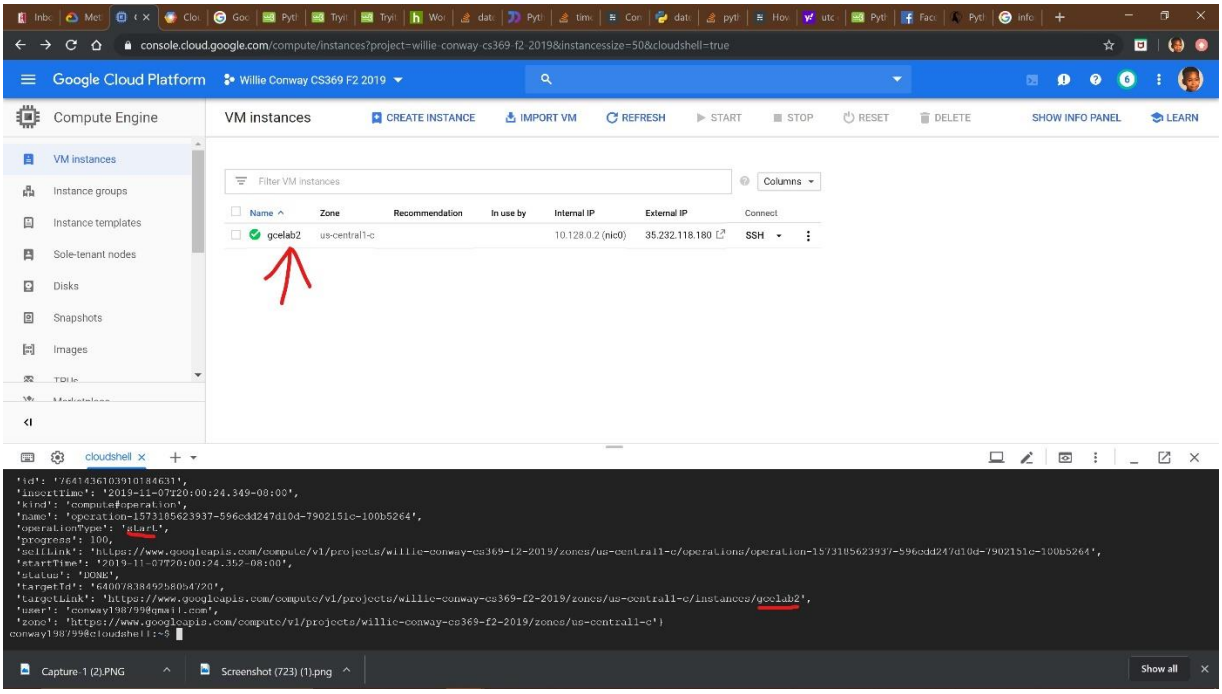
(Output of the User name, Operation(stop), Starting date, and Starting time. I was able to obtain this by selecting specific elements from the response dictionary. See code below.)



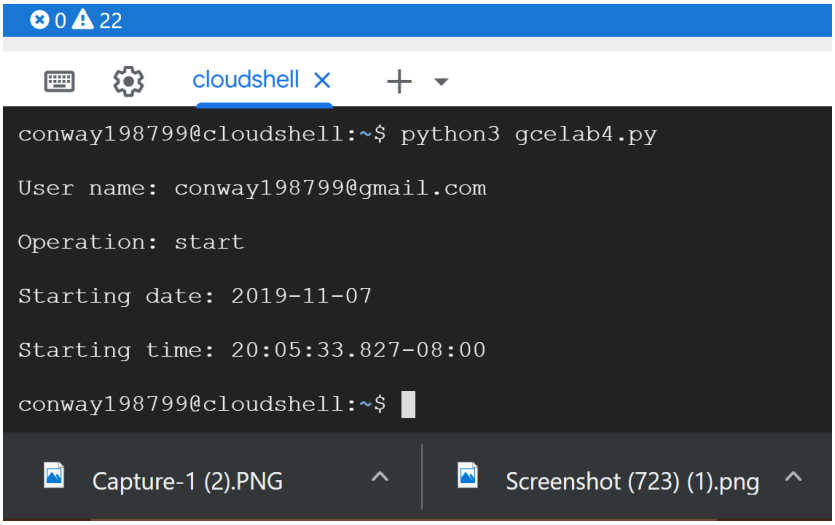
(The following code above displays the following output of the screenshot above the code. To obtain elements form the response dictionary, I had to go back to the previous output to see the selected elements and how they were typed, because that would be the only way to get my code to react to the previous code that was copied. In the last #TODO section I used request as a string to process the response['startTime']. Due to the response['startTime'] having one line of the date and time, I needed to use a fuction to split both, so that each would print on a separate line. To do this I used the slice() function, which allows my to slice characters from and object and return the slice object. So in the first string I use slice(10) to slice the time off of response['startTime'], then I used the variable (x) to return the first sliced object. For the second string I used slice(11,29) to slice the date from response['startTime'], then used the variable (a) to return the sliced object. For the remaining elements,

response['user'] and respone['operationType'], all I had to do was print out the following elements. This then gave me the first screenshot of exercise 5, after running the program gcelab3.py.)

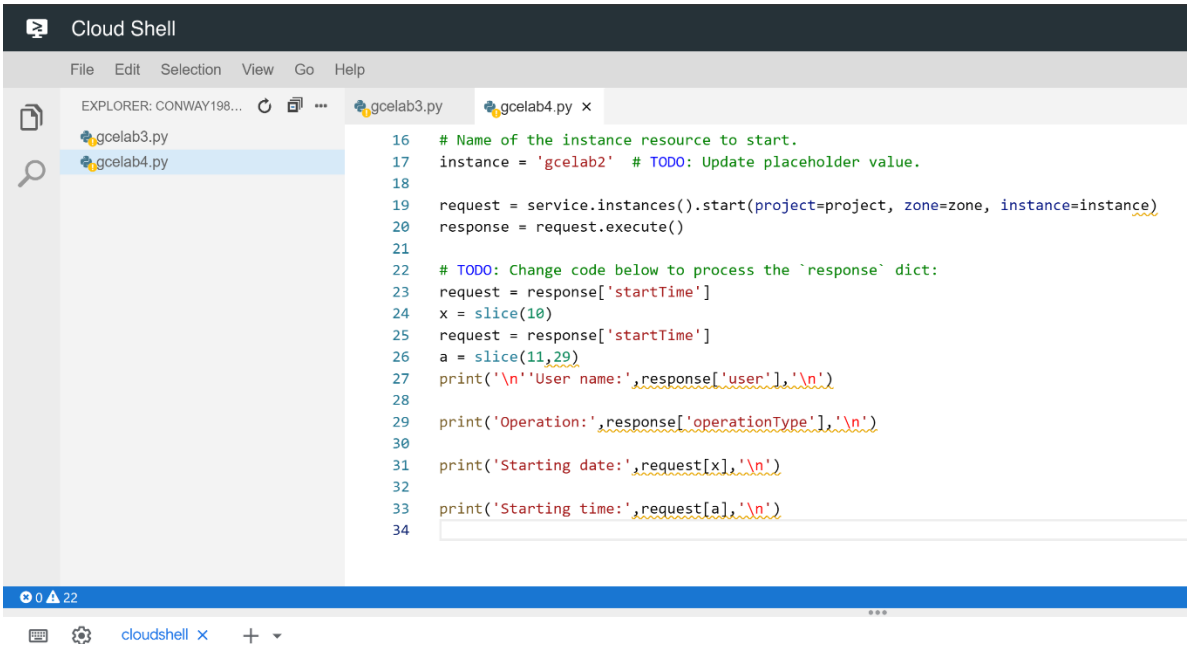
6. In addition to stopping your instance, you should now be able to write a python program to start your instance. See the page Method: instances.start (Links to an external site.)for all the details, though you should find things are very similar to what you have just done. This program should print the same type of information as in step 5 of this assignment. Take a screenshot of its output.



(This is an output to show that I entered all the following items correctly to get the same output from the website, of the complete response dictionary. It also showed that my instance gcelab2 has completely started. To activate this command, I entered python3 gcelab4.py. This tells the Cloud Shell that I'm using python version 3 and that I want to open the file gcelab4.py.)



(Output of the User name, Operation(start), Starting date, and Starting time. I was able to obtain this by selecting specific elements from the response dictionary. See code below.)



(The same fundamentals that were applied in step 5 are applied in step 6. So, I applied the same code to get the similar output.Then processed the program using python gcelab4.py.)

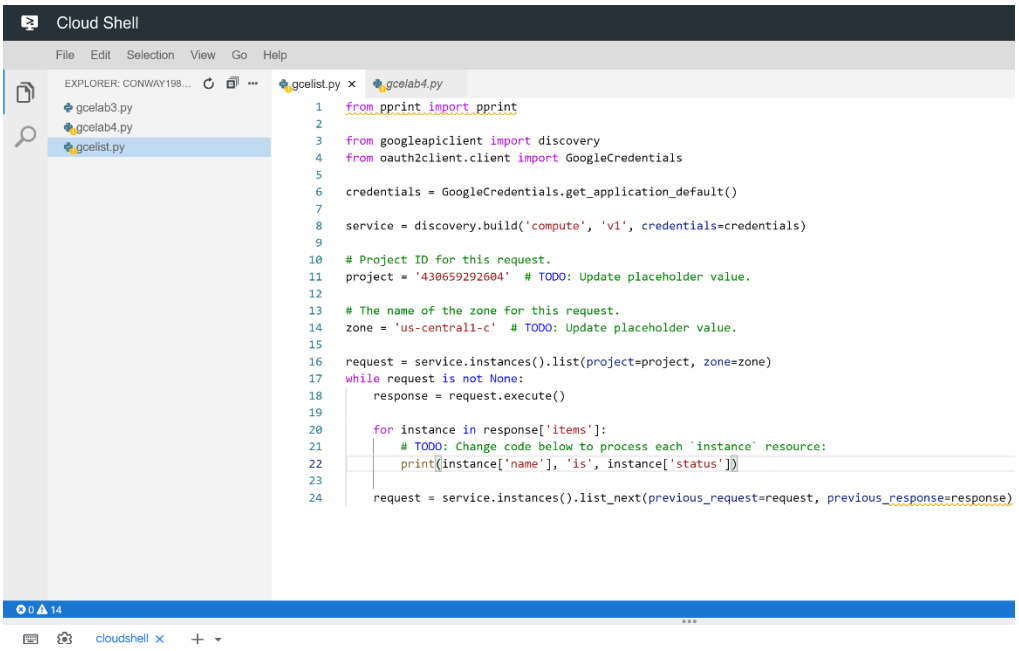
7. In your Word document, write a paragraph about what you did to write the program in step 6. Include any Linux commands you used.

   (see Summary2.docx)

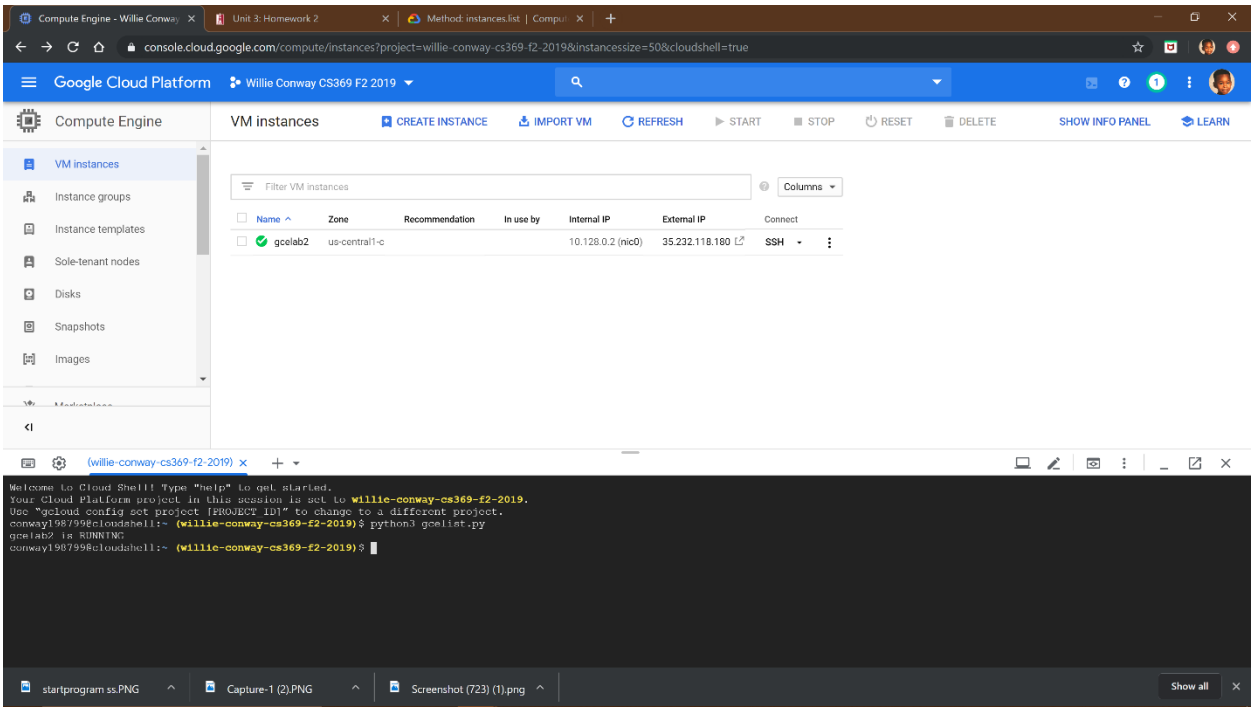
8. See the page Method: instances.list (Links to an external site.), for generating a list of your instances. Copy the python program and upload it to your Cloud Shell. Edit the program to specify your project and zone, then replace the print statement with

print(instance['**name**'], '**is**', instance['**status**'])

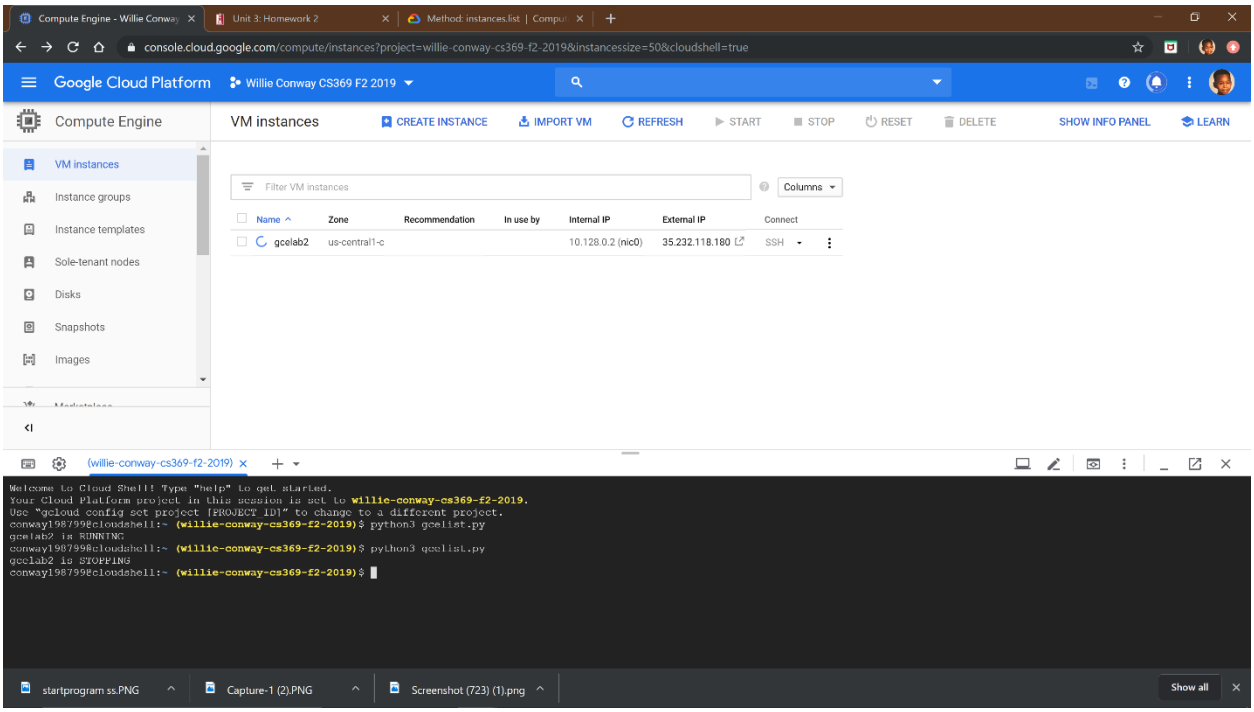
Run your program to get a list of your instances and their current status.



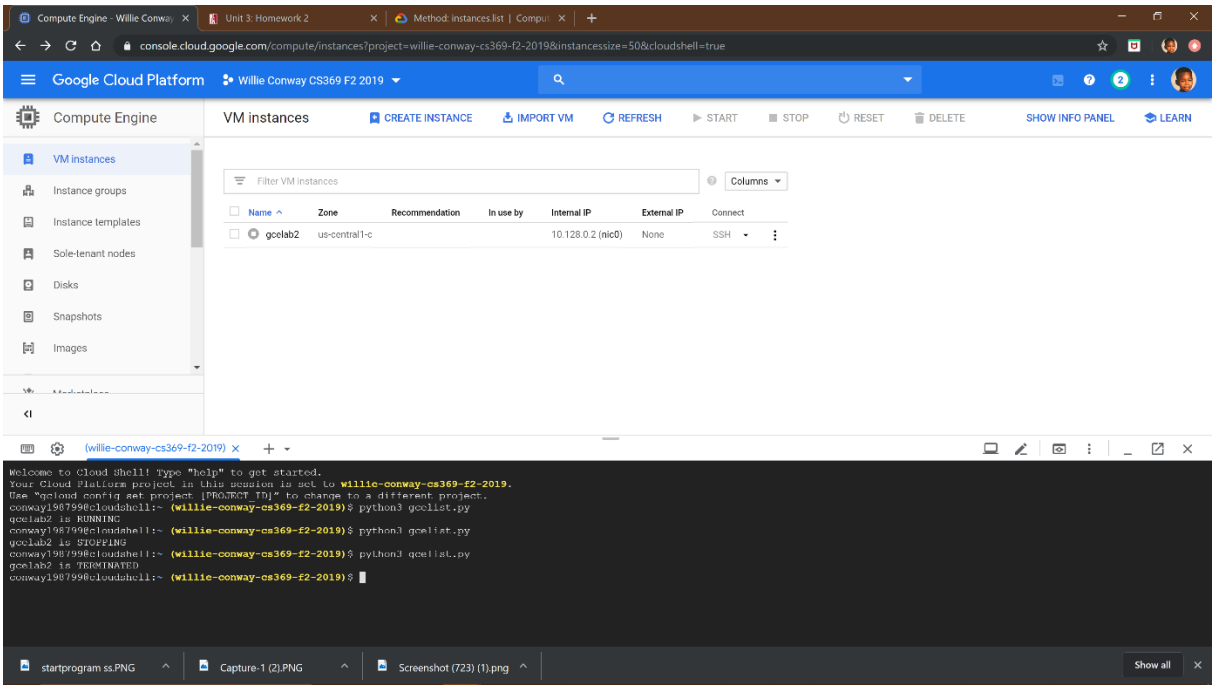
(Created file gcelist.py and copy pasted the code from Method: instances.list (Links to an external site.) "Examples" section. Then I updated pprint(instance) to print(instance['name'], 'is', instance['status']). This code is used for generating a list of instances.)



(Output of showing the list of instances that are their current status. Since I only have one instance, gcelab2 is in the current running state.)



(Output of showing the list of instances that are their current status. Since I only have one instance, gcelab2 is in the current stopping state.)

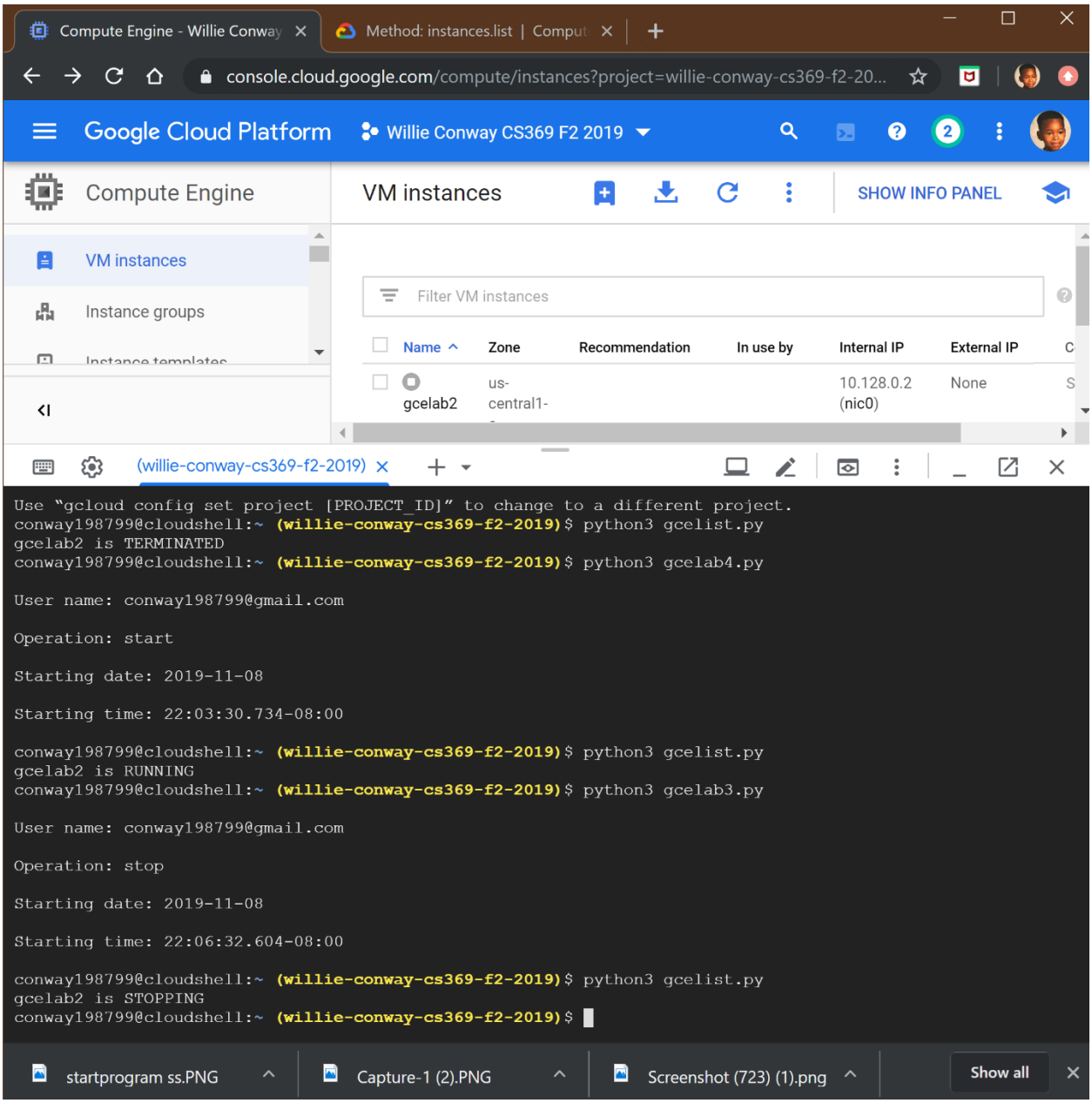


(Output of showing the list of instances that are their current status. Since I only have one instance, gcelab2 is in the current terminated state.)

9. Stop your instance, any way that you like. Then run your programs in this order

list, start (wait for it to finish), list, stop, list

Take a screenshot of the output, including the command prompt, so that it shows the following information. Be sure it is a screenshot image.



(Outputs the programs in the order of list, start (wait for it to finish), list, stop, list in functioning the gcelab2 instance.)