**Cheng, Loi**

9/12/2019

CSCIE-88, 2019 Fall

Homework 1: AWS and Docker Setup

This document is a template for your solutions submission. You are free to add additional information in this submission if you would like. Extra screenshots and extra documentation is perfectly fine. Screenshots must always be viewable. If a screenshot is too blurry to be viewed or is chopped off in a key area you will not receive full credit for it.

Please identify which problems were completed. If any were incomplete, please identify where you encountered problems.

|  |
| --- |
| *for example:*  Problem 1: 100% complete  Problem 2: 100% complete  Problem 3: 100% complete  Problem 4: 100% complete  Problem 5 Bonus: 100% complete |

**Problem 1: [25 points] File generator program**

Paste your source code into the following area. All code should be heavily commented, and easily readable. [15 points]

|  |
| --- |
| import random  import threading  import argparse  def makeFile(fileName, numLines):  '''  generate a file with the specified number of lines:  each line should have 3 random numbers in the range [0-10]; the lines would look like:  1 7 3  2 1 3  '''  # open a (new) file to write  outF = open(fileName, "w")  for line in range (numLines):  # write line to output file  outF.write( str(random.randint(1,10)) + " " + str(random.randint(1,10)) + " " + str(random.randint(1,10)) )  outF.write("\n")  outF.close()  def makeManyFiles(numFiles, numLines, filePrefix):  '''  generate specified number of files with the specified number of lines:  each line should have 3 random numbers in the range [0-10]; the lines would look like:  1 7 3  2 1 3    File names are in the format: <yourFirstName>\_<yourLastName>\_threadNumber.txt  For example, if 'numFiles' = 3, the following files should be generated:  marina\_popova\_0.txt  marina\_popova\_1.txt  marina\_popova\_2.txt  '''  for fileCount in range (numFiles):    fileName = filePrefix + str(fileCount) + ".txt"  #run thread  threading.Thread(target=makeFile, args=(fileName,numLines)).start()  def main():  # get input arguments  parser = argparse.ArgumentParser(description='hw1\_problem1')  parser.add\_argument('numFiles', type=int, help='number of files')  parser.add\_argument('numLines', type=int, help='number of lines')  args = parser.parse\_args()  print(args)  # make many files  makeManyFiles(args.numFiles, args.numLines, "loi\_cheng\_")  if \_\_name\_\_ == "\_\_main\_\_":  main() |

Paste an example of your code output into the following area. This can be a screenshot (ideally), or a copy/paste of console text. [5 points]

|  |
| --- |
|  |

Paste an example of the contents of one of your generated files in the following area. [5 points]

|  |
| --- |
| 4 1 3  8 3 4  6 5 8  9 2 9  7 9 9  8 2 3  6 3 9  3 5 4  4 9 2  5 2 7 |

**Problem 2: [25 points] Set up a machine and demonstrate that it works**

Paste a screenshot of your machine, include your owner information and creation date in your screenshot. [15 points]

|  |
| --- |
| Example:    Screenshot: |

Describe how you connected to your machine:

|  |
| --- |
| I connected to it through ssh |

Show which Java and/or Python version is installed on your machine:

|  |
| --- |
| Example:    Screenshot: |

Paste a screenshot of the command you used to transfer your program to your machine [5 points]

|  |
| --- |
|  |

Paste a screenshot of your program execution from within your machine. [5 points]

|  |
| --- |
|  |

**Problem 3: [25 points] Run Redis server and clients as Docker containers and demonstrate that they work**

Show all the commands you used, in sequence, to start your Redis server and clients [15 points]

|  |
| --- |
| Example:    Screenshot: |

Show the value of ‘x’ in the clients, as described in problem 3 [10 points]

|  |
| --- |
| Server    Client 1    Client 2    Client 1 |

**Problem 4: [25 points] Run Postgres DB as Docker container and demonstrate that it works**

Show all the commands you used, in sequence, to start your Postgres server. [10 points]

|  |
| --- |
| Example:    Screenshot: |

Show how you connect to the DB [5 points]

|  |
| --- |
| I connected to the DB using Dbeaver GUI    The connection was made with the specified password    A table was made in the DB with the features as shown |

Show results of querying your database for all records. [10 points]

|  |
| --- |
| A SQL query was done to show all the records |

**Problem 5: [Bonus, 15 points]: Start multiple Docker container via Compose**

Show your Docker Compose configuration [7 points]

|  |
| --- |
|  |

Show that theRedis server, 2 Redis clients, Postgres server are all functional [8 points]

|  |
| --- |
| All docker containers are running according to the docker-compose.yml file    The redis server ip address is retreived    Redis client one is logged in to the server, and a key x is created and set to 10    Redis client two is logged in to the server, and the key x is retrieved as 10  The key x is changed to 20    Redis client one retrieves the key x again and is 20    Using Dbeaver, I logged into the postgres server specifed in the docker-compose.yml file    The connection to the server was successful |