**Bates, Ethan**

9/9/2019

CSCI E-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.*

|  |
| --- |
| 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 os  import sys  import threading  # create a string of 3 random numbers 1-10  def gen\_three\_random\_nums():  i = 1  numbers = []  while i <= 3:  new\_num = random.randint(0, 10)  numbers.append(new\_num)  i += 1  num\_string=str(numbers[0]) + " " + str(numbers[1]) + " " + str(numbers[2]) +'\n'  return num\_string  # create file with X number of lines  def gen\_ran\_num\_file(number\_lines,file\_id):  file\_name = f"ethan\_bates\_{file\_id}.txt"  export\_file = os.path.join(os.getcwd(), file\_name)  try:  input\_num = int(number\_lines)  except ValueError:  print(f"[Error] {number\_lines} not a valid number. Try again...")  with open(export\_file,'w') as out\_file:  ln=1  while ln <= input\_num:  random\_num\_line = gen\_three\_random\_nums()  out\_file.write(random\_num\_line)  ln +=1  out\_file.close()  print(f"Wrote [{number\_lines}] lines to [{file\_name}]")  # open thread and call file generator  def thread\_worker\_function(input\_num\_lines,input\_thread\_id):  print(threading.current\_thread().getName(), 'Starting')  gen\_ran\_num\_file(input\_num\_lines,input\_thread\_id)  print(threading.current\_thread().getName(), 'Exiting')  # main application call  if \_\_name\_\_ == '\_\_main\_\_':  if len(sys.argv) < 3:  print("You must input 2 parameters: [numFiles] [numLines]")  exit  else:  numFiles = sys.argv[1]  numLines = sys.argv[2]    # generate files  print(f"Generating [{numFiles}] files using [{numFiles}] threads with [{numLines}] lines each")  threads = []  for i in range(int(numFiles)):  t = threading.Thread(target=thread\_worker\_function, args=(numLines,i+1,))  threads.append(t)  t.start() |

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]

|  |
| --- |
| -- Python program takes two system input params (numFiles, numLines) |

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

|  |
| --- |
| -- ls show files, cat shows contents, wc -l shows number of lines |

**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]

|  |
| --- |
|  |

Describe how you connected to your machine:

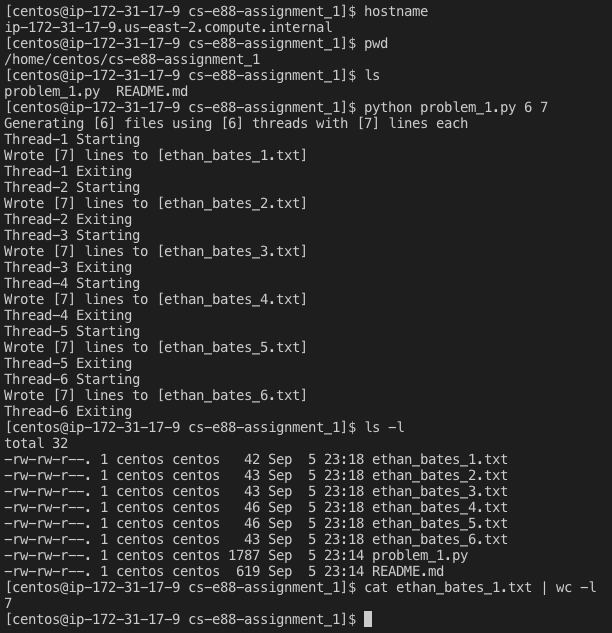
|  |
| --- |
| 1. Set Security Group to Allow SSH from my IP 2. Launched EC2 and create key pair file 3. SSHed into machine    1. ***ssh -i "harvard-e88-hw1-ebb.pem" centos@ec2-18-219-207-90.us-east-2.compute.amazonaws.com*** |

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

|  |
| --- |
| -- Initial:    -- Also installed python3 to run program: |

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

|  |
| --- |
| -- I create a git repo; pushed my code; cloned the repo on the EC2 |

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]

|  |
| --- |
| -- started redis-server, and 2 redis clients (aptly named) |

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

|  |
| --- |
| -- redis-server is the host, set value of x in redis-client1, then changed in redis-client2 |

**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]

|  |
| --- |
| Pulled postgres docker image (postgres 11 is latest), started with port forwarding |

Show how you connect to the DB [5 points]

|  |
| --- |
| -- exec into container, run psql as postgres user to postgres db |

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

|  |
| --- |
| -- create table and show definition    -- insert values, select them, show there are 3, then delete them |

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

Show your Docker Compose configuration [7 points]

|  |
| --- |
| **version: '3'**  **services:**  **redis:**  **container\_name: redis-server**  **image: redis**  **ports:**  **- "6379:6379"**  **redis-client1:**  **container\_name: redis-client1**  **image: redis**  **links:**  **- redis**  **redis-client2:**  **container\_name: redis-client2**  **image: redis**  **links:**  **- redis**  **postgres:**  **container\_name: postgres-server**  **image: postgres**  **ports:**  **- "5432:5432"** |

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

|  |
| --- |
| -- no containers, ran docker-compose, all containers up and running    -- able to connect to redis server and redis clients    -- able to connect to postgres |