

**Carlebach, Mark**  
**September 4, 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: 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]

```
"""
I watched Youtube video by Derek Banas "Learn to Program 14: Python Threads"
to learn how to use threads in Python for this assignment.
"""

# Import needed libraries
import random
import threading
import argparse

# Set random seed for repeatability
random.seed(88)

# input arguments
# Example of running: python problem1.py --numLines 3 --numFiles 4
parser = argparse.ArgumentParser()
parser.add_argument("--numFiles", help="Number of files to create")
parser.add_argument("--numLines", help="Number of lines in each file")
args = parser.parse_args()
numFiles = int(args.numFiles)
numLines = int(args.numLines)
```

```

def exec_thread(thread_num, numLines):
    """On a thread, create file with 3 random ints on each of numLines
    :thread_num (int): the thread number
    :numLines (int): the number of lines to write in file
    :returns None, side effects only
    """

    # Create list numLines long with each item being a 3 random ints
    lines = [f"{random.randint(0, 10)} " + \
             f"{random.randint(0, 10)} " + \
             f"{random.randint(0, 10)}\n" for i in range(numLines)]

    # Open a file with name that references thread_num
    with open(f"mark_carlebach_{thread_num}.txt", mode="w") as f:
        # Write lines to file
        f.writelines(lines)

# Echo system arguments
print(f"Creating {numFiles} files with {numLines} lines per file...")

# For each file/thread...
for i in range(numFiles):

    # Create thread to execute function that creates file
    thread = threading.Thread(target=exec_thread,
                              args=(i, numLines))

    # Start the thread
    thread.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]

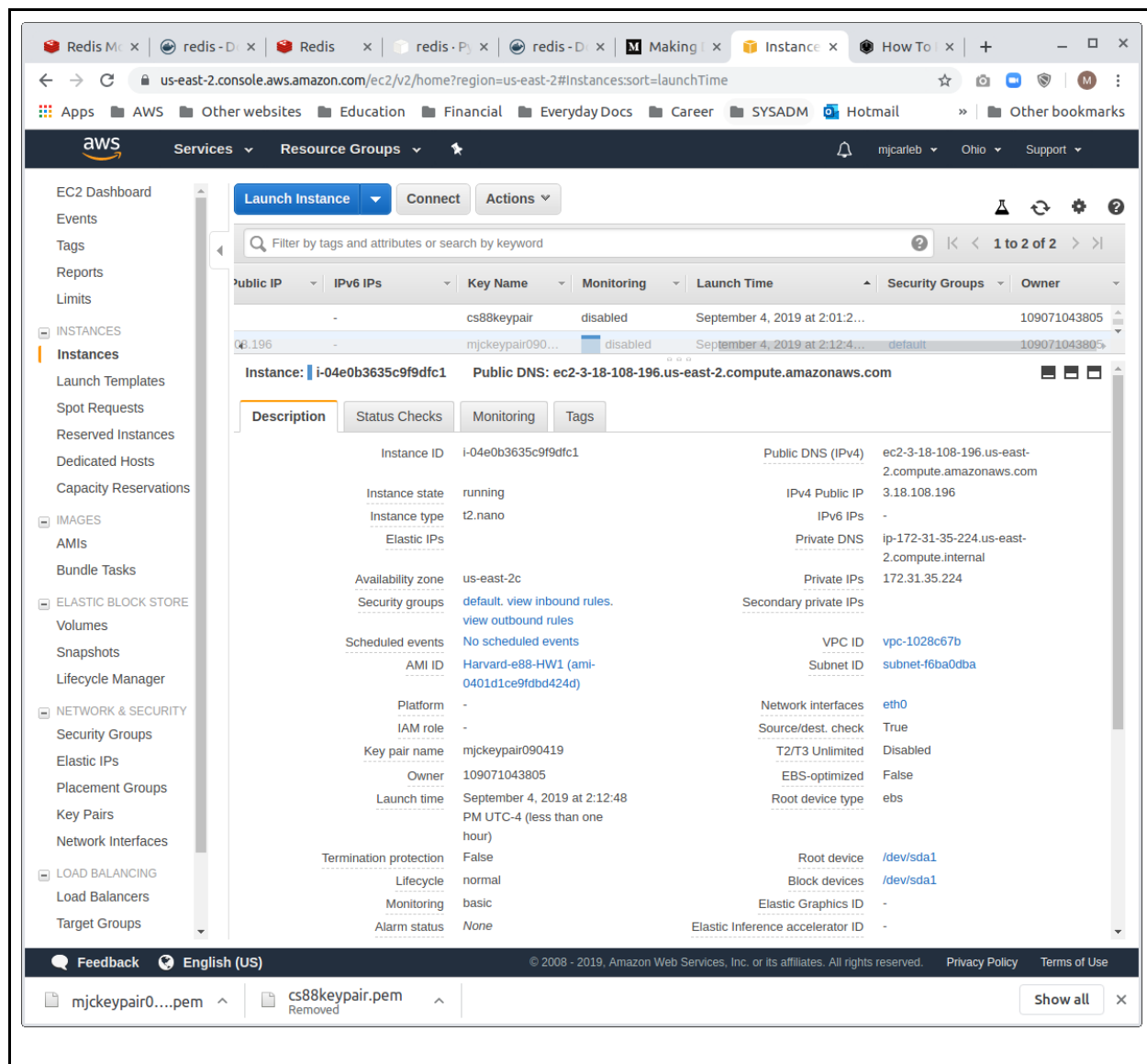
```
mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(cs88) mjcarleb@omen:~/HES/cs88/HW1$ python problem1.py --numFiles 5 --numLines 6
Creating 5 files with 6 lines per file...
(cs88) mjcarleb@omen:~/HES/cs88/HW1$
```

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

```
mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(cs88) mjcarleb@omen:~/HES/cs88/HW1$ cat mark_carlebach_4.txt
7 5 9
5 9 9
9 6 2
5 4 10
6 8 9
5 6 7
(cs88) mjcarleb@omen:~/HES/cs88/HW1$
```

**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:

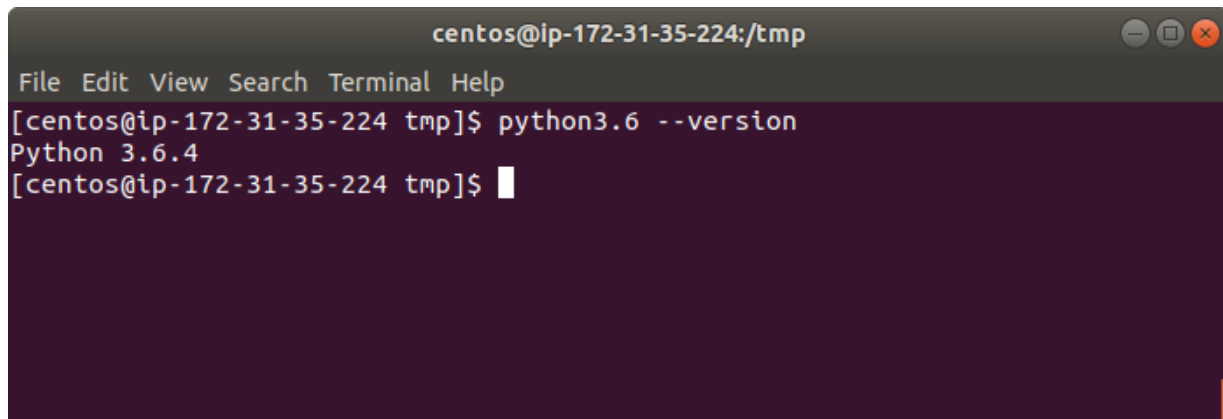
```
$ ssh -i "~/mjckeypair090419.pem" centos@ec2-3-18-108-196.us-east-2.compute.amazonaws.com
```

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

I saw the machine had Python 2.7 installed. So, I upgraded to 3.6 using the following commands (from this reference: <https://www.rosehosting.com/blog/how-to-install-python-3-6-4-on-centos-7/>):

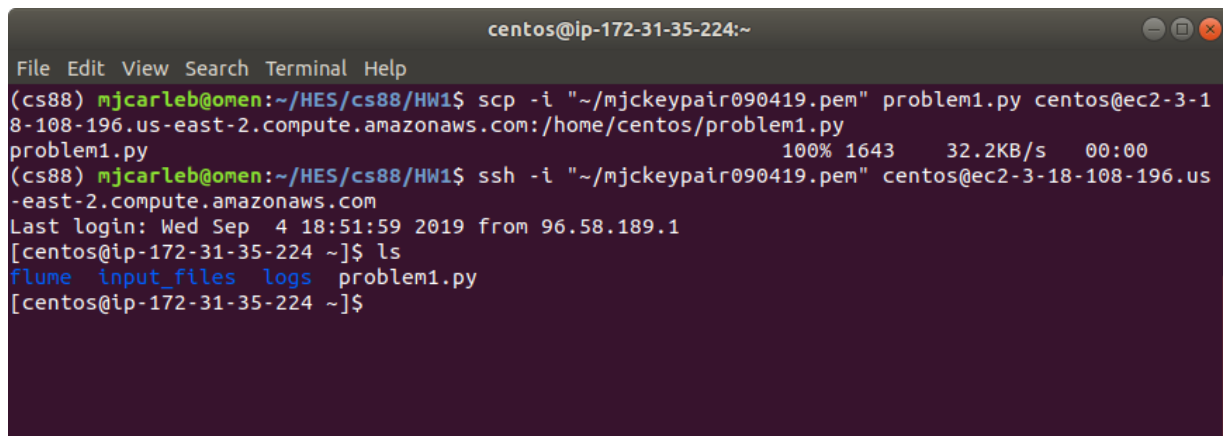
```
sudo yum install -y https://centos7.iuscommunity.org/ius-release.rpm
sudo yum update
sudo yum install -y python36u python36u-libs python36u-devel python36u-pip
```

I now, I have Python 3.6 as shown in the screenshot below:

A terminal window titled 'centos@ip-172-31-35-224:/tmp' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'python3.6 --version' being executed, which returns 'Python 3.6.4'. The prompt is '[centos@ip-172-31-35-224 tmp]\$'.

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

```
scp -i "~/mjckeypair090419.pem" problem1.py centos@ec2-3-18-108-196.us-east-2.compute.amazonaws.com:/home/centos/problem1.py
```

A terminal window titled 'centos@ip-172-31-35-224:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a user 'mjcarleb@omen' in a directory '~/HES/cs88/HW1' executing an 'scp' command to transfer 'problem1.py' to a remote server. The command is: 'scp -i "~/mjckeypair090419.pem" problem1.py centos@ec2-3-18-108-196.us-east-2.compute.amazonaws.com:/home/centos/problem1.py'. The output shows '100% 1643 32.2KB/s 00:00'. Then, the user executes 'ssh -i "~/mjckeypair090419.pem" centos@ec2-3-18-108-196.us-east-2.compute.amazonaws.com'. The terminal shows the login message 'Last login: Wed Sep 4 18:51:59 2019 from 96.58.189.1' and the prompt '[centos@ip-172-31-35-224 ~]\$'. Finally, the user executes 'ls', which lists 'flume input\_files logs problem1.py'. The prompt is '[centos@ip-172-31-35-224 ~]\$'.

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

```
centos@ip-172-31-35-224:~  
File Edit View Search Terminal Help  
[centos@ip-172-31-35-224 ~]$ python3.6 problem1.py --numFiles 5 --numLines 6  
Creating 5 files with 6 lines per file...  
[centos@ip-172-31-35-224 ~]$
```

**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]

```
docker network create my_network  
  
docker run --name my_redis_server -d --network my_network redis redis-server --appendonly yes  
  
docker run --name my_redis_client1 -itd --network my_network redis  
  
docker run --name my_redis_client2 -itd --network my_network redis
```

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

Working inside my\_redis\_client1:

```
mjcarleb@omen: ~/HES/cs88/HW1  
File Edit View Search Terminal Help  
(base) mjcarleb@omen:~/HES/cs88/HW1$ docker exec -it my_redis_client1 sh  
# redis-cli  
127.0.0.1:6379> connect my_redis_server 6379  
my_redis_server:6379> SET x 10  
OK  
my_redis_server:6379> GET x  
"20"  
my_redis_server:6379>
```

Working inside my\_redis\_client2:

```
mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(base) mjcarleb@omen:~/HES/cs88/HW1$ docker exec -it my_redis_client2 sh
# redis-cli
127.0.0.1:6379> connect my_redis_server 6379
my_redis_server:6379> GET x
"10"
my_redis_server:6379> SET x 20
OK
my_redis_server:6379> 
```

**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]

```
docker network create my_network

docker run --network my_network --name my_postgres_server -e
POSTGRES_PASSWORD=first1 -d postgres
```

Show how you connect to the DB [5 points]

```
docker run -it --rm --network my_network postgres psql -h my_postgres_server -U postgres
```

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

**You did not ask, but here are psql commands to create database:**

- CREATE TABLE carlebach\_data (ID serial NOT NULL PRIMARY KEY, name text NOT NULL, creation\_date date);
- INSERT INTO carlebach\_data (name, creation\_date) VALUES ('Shea Patterson', '9/5/2019'), ('Ambry Thomas', '9/5/2019'), ('Mazie Smith', '9/5/2018');
- SELECT \* FROM carlebach\_data;
- delete from carlebach\_data \*;

**Here are query results:**

id	name	creation_date
1	Shea Patterson	2019-09-05
2	Ambry Thomas	2019-09-05
3	Mazie Smith	2018-09-05

```
mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(cs88) mjcarleb@omen:~/HES/cs88/HW1$ docker run --network my_network --name my_postgres_server -e POSTGRES_PASSWORD=first1 -d postgres
44a2160dfa02d702c82f9806daa497a2748ff0dbddbc4388686071d1cbbdie188
(cs88) mjcarleb@omen:~/HES/cs88/HW1$ docker run -it --rm --network my_network postgres psql -h my_postgres_server -U postgres
Password for user postgres:
psql (11.5 (Debian 11.5-1.pgdg90+1))
Type "help" for help.

postgres=# CREATE TABLE carlebach_data (ID serial NOT NULL PRIMARY KEY, name text NOT NULL, creation_date date);
CREATE TABLE
postgres=# INSERT INTO carlebach_data (name, creation_date) VALUES ('Shea Patterson', '9/5/2019'), ('Ambry Thomas', '9/5/2019'), ('Mazie Smith', '9/5/2018');
INSERT 0 3
postgres=# SELECT * FROM carlebach_data;
 id |      name      | creation_date
-----+-----+-----
  1 | Shea Patterson | 2019-09-05
  2 | Ambry Thomas  | 2019-09-05
  3 | Mazie Smith   | 2018-09-05
(3 rows)

postgres=#
```

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

Show your Docker Compose configuration [7 points]

```
mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(base) mjcarleb@omen:~/HES/cs88/HW1$ cat docker-compose.yml
version: '3'
services:
  my_redis_server:
    image: redis
    command:
      redis-server
  my_redis_client1:
    image: redis
  my_redis_client2:
    image: redis
  my_postgres_server:
    image: postgres
    environment:
      - POSTGRES_PASSWORD=first1
(base) mjcarleb@omen:~/HES/cs88/HW1$
```

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

Show all servers running with `$docker ps`:



```

mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(base) mjcarleb@omen:~/HES/cs88/HW1$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
1b43fc9d757f   redis     "docker-entrypoint.s..." 14 seconds ago Up 12 seconds 6379/tcp                hw1_my_redis_client1_1
87820bb8d7d5   redis     "docker-entrypoint.s..." 14 seconds ago Up 13 seconds 6379/tcp                hw1_my_redis_server_1
6cedbb97e55e   postgres  "docker-entrypoint.s..." 14 seconds ago Up 13 seconds 5432/tcp                hw1_my_postgres_server_1
ce8fb31e62b3   redis     "docker-entrypoint.s..." 14 seconds ago Up 13 seconds 6379/tcp                hw1_my_redis_client2_1
(base) mjcarleb@omen:~/HES/cs88/HW1$

```

Get network name created by docker-compose with `$docker network list`:

```

mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(base) mjcarleb@omen:~/HES/cs88/HW1$ docker network list
NETWORK ID      NAME                DRIVER              SCOPE
b7aba16a7551    bridge              bridge              local
f49b7e972cfe    host                host                local
338fe5b44188    hw1_default         bridge              local
f035951b600a    none                null                local
(base) mjcarleb@omen:~/HES/cs88/HW1$

```

Connect to postgres server with `$docker run -it --rm --network hw1_default postgres psql -h hw1_my_postgres_server_1 -U postgres`:

```

mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(base) mjcarleb@omen:~/HES/cs88/HW1$ docker run -it --rm --network hw1_default postgres psql -h hw1_my_postgres_server_1 -U postgres
Password for user postgres:
psql (11.5 (Debian 11.5-1.pgdg90+1))
Type "help" for help.

postgres=#

```

From `redis_client1`, update `redis_server` with `$docker exec -it...`:

```

mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(base) mjcarleb@omen:~/HES/cs88/HW1$ docker exec -it hw1_my_redis_client1_1 sh
# redis-cli
127.0.0.1:6379> connect hw1_my_redis_server_1 6379
hw1_my_redis_server_1:6379> SET x 10
OK
hw1_my_redis_server_1:6379> GET x
"10"
hw1_my_redis_server_1:6379>

```

From redis\_client2, read value from redis\_server (that was created by redis\_client1):

```
mjcarleb@omen: ~/HES/cs88/HW1
File Edit View Search Terminal Help
(base) mjcarleb@omen:~/HES/cs88/HW1$ docker exec -it hw1_my_redis_client2_1 sh
# redis-cli
127.0.0.1:6379> connect hw1_my_redis_server_1 6379
hw1_my_redis_server_1:6379> GET x
"10"
hw1_my_redis_server_1:6379> 
```