Chaparro, Alejandro
09-10-2019
CSCIE-88, 2019 Fall

Homework 1: AWS and Docker Setup

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

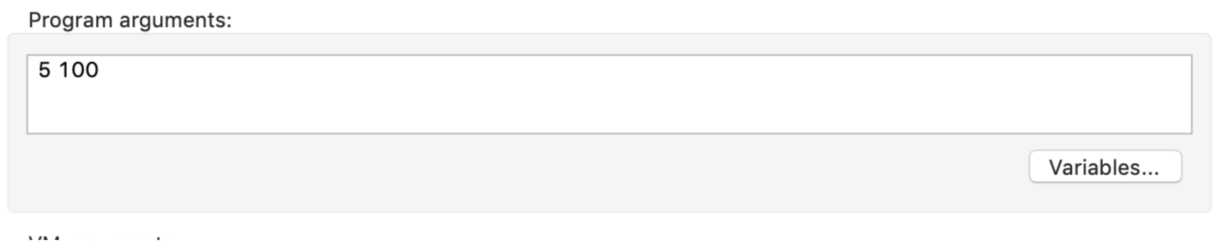
Problem 1: [25 points] File generator program

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

```
package edu.harvard.fall2019.cscie88.acg.hw1;

public class Main {

        public static void main(String[] args) {

                //If number of parameters is less than 2, the program is not executed
                if(args.length >= 2) {

                        //Number of files to be generated
                        int numFiles = Integer.parseInt(args[0]);

                        //Number of lines to be generated into each file
                        int numLines = Integer.parseInt(args[1]);

                        //Up to numFiles threads are instantiated and run.
                        //Each thread creates a file.
                        for(int i = 0; i < numFiles; i++) {
                                new Thread(new FileGenerator(i, numLines)).start();;
                        }
                }

        }
}
```

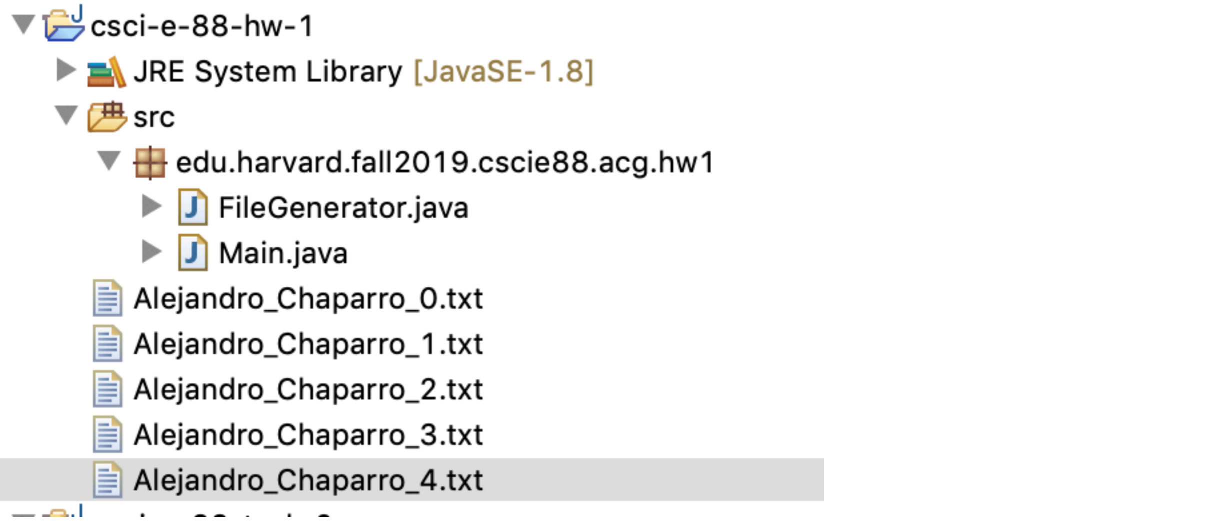
```java
package edu.harvard.fall2019.cscie88.acg.hw1;

import java.io.BufferedWriter;
import java.io.IOException;
import java.nio.charset.Charset;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

/**
 * FileGenerator creates a single file with numLines number of lines.
 * It implements Runnable so that it can be executed within a Thread.
 */
public class FileGenerator implements Runnable{

        private int threadNumber; //Id of the file.  It ranges from 0 to numFiles.
        private int numLines;   //Number of lines to be generated into the file.

        public FileGenerator(int threadNumber, int numLines) {
                this.threadNumber = threadNumber;
                this.numLines = numLines;
        }

        @Override
        public void run() {

                //The files are generated on the same directory
                //where the program is run from
                Path path = Paths.get("./", String.format(
                                "Alejandro_Chaparro_%d.txt", threadNumber));

                try(BufferedWriter writer = Files.newBufferedWriter(
                                path, Charset.forName("UTF-8"))){

                        //Line separator is initially an empty string and then
                        //it becomes a change of line after the first line
                        //is added to the file
                        String lineSeparator = "";

                        for(int c = 0; c < numLines; c++) {

                                //A line will contain max 9 characters.
                                //Therefore StringBuilder is instantiated with 9
                                //characters of capacity
                                StringBuilder lineBuilder = new StringBuilder(8);
                                String separator = "";
                                lineBuilder.append(lineSeparator);
                                lineSeparator = "\n";

                                //Three random numbers between 0 and 10
                                //are generated and appended to the line
                                for(int i = 0; i < 3; i++) {
                                        lineBuilder.append(separator);
                                        lineBuilder.append((int)(Math.random() * 11));
                                        separator = " ";
                                }

                                //The line with the three random numbers is written to the file
                                writer.write(lineBuilder.toString());
                        }

                }catch(IOException ex){
                        ex.printStackTrace();
                }
        }
}
```

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]



Program arguments:

```
5 100
```



```
▼ csci-e-88-hw-1
    ▶ JRE System Library [JavaSE-1.8]
    ▼ src
        ▼ edu.harvard.fall2019.cscie88.acg.hw1
            ▶ FileGenerator.java
            ▶ Main.java
      Alejandro_Chaparro_0.txt
      Alejandro_Chaparro_1.txt
      Alejandro_Chaparro_2.txt
      Alejandro_Chaparro_3.txt
      Alejandro_Chaparro_4.txt
```

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

```
9 10 4
8 10 6
2 7 6
10 5 9
4 1 6
2 7 1
9 4 3
6 5 0
7 9 10
7 7 9
10 8 1
0 6 1
4 1 8
4 2 2
```

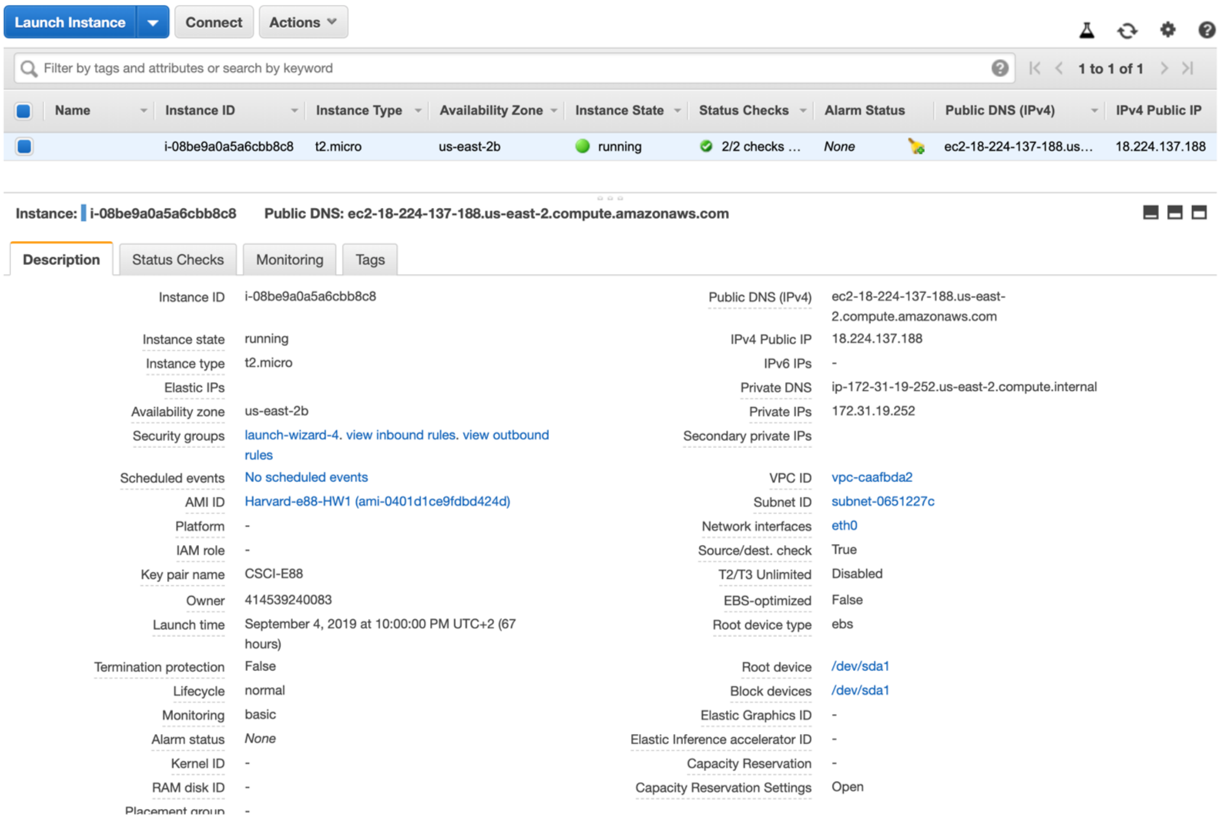
```
4 9 0
8 0 4
9 5 6
0 0 9
9 5 7
9 6 4
0 10 3
1 6 9
10 6 10
9 6 7
2 9 0
7 7 0
9 8 9
10 3 7
8 7 2
7 9 8
9 9 1
4 1 6
6 10 3
1 4 1
8 10 6
7 1 4
10 10 9
7 2 0
9 3 1
2 0 9
5 3 4
6 10 3
1 4 9
4 5 3
10 0 6
8 10 3
9 4 9
10 3 7
0 6 10
1 6 10
0 4 8
0 1 5
1 0 7
1 7 6
10 8 0
7 4 2
0 8 5
8 3 10
1 5 9
7 5 1
6 10 1
2 10 1
3 5 2
3 10 2
4 6 4
1 6 5
6 5 0
3 4 10
1 2 6
0 5 2
9 0 5
1 8 8
3 0 6
```

```
3 9 7
3 1 5
9 4 1
10 0 6
10 6 6
10 4 4
5 0 3
9 8 8
4 1 9
7 9 1
1 9 10
2 8 2
7 8 2
2 8 10
0 9 1
3 1 10
7 7 7
1 2 4
8 10 3
1 2 8
0 5 1
1 4 10
8 1 5
6 5 3
7 8 9
6 4 5
6 1 10
```

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]

Screenshot:



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-08be9a0a5a6cbb8c8	t2.micro	us-east-2b	● running	✓ 2/2 checks ...	None	ec2-18-224-137-188.us...	18.224.137.188

Instance: i-08be9a0a5a6cbb8c8 **Public DNS:** ec2-18-224-137-188.us-east-2.compute.amazonaws.com

Instance ID	i-08be9a0a5a6cbb8c8	Public DNS (IPv4)	ec2-18-224-137-188.us-east-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	18.224.137.188
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-19-252.us-east-2.compute.internal
Availability zone	us-east-2b	Private IPs	172.31.19.252
Security groups	launch-wizard-4. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-caafbda2
AMI ID	Harvard-e88-HW1 (ami-0401d1ce9fdbd424d)	Subnet ID	subnet-0651227c
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	CSCI-E88	T2/T3 Unlimited	Disabled
Owner	414539240083	EBS-optimized	False
Launch time	September 4, 2019 at 10:00:00 PM UTC+2 (67 hours)	Root device type	ebs
Termination protection	False	Root device	/dev/sda1
Lifecycle	normal	Block devices	/dev/sda1
Monitoring	basic	Elastic Graphics ID	-
Alarm status	None	Elastic Inference accelerator ID	-
Kernel ID	-	Capacity Reservation	-
RAM disk ID	-	Capacity Reservation Settings	Open
Placement group	-		

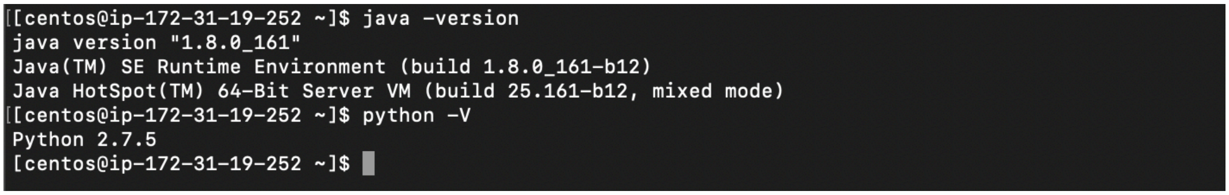
Describe how you connected to your machine:

I have created a new key pair called CSCI-E88 and downloaded the .pem file. Then I've used this file to connect to the instance through ssh by issuing the next command:

```
ssh -i "CSCI-E88.pem" centos@ec2-18-224-137-188.us-east-
2.compute.amazonaws.com
```

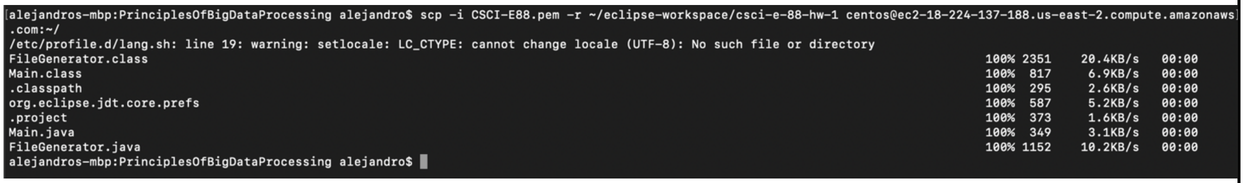
Show which Java and/or Python version is installed on your machine:

Screenshot:



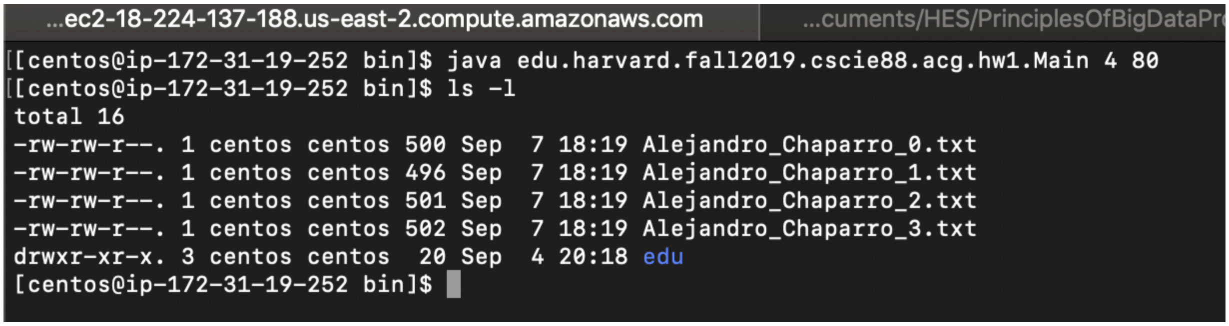
```
[[centos@ip-172-31-19-252 ~]$ java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
[[centos@ip-172-31-19-252 ~]$ python -V
Python 2.7.5
[centos@ip-172-31-19-252 ~]$
```

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



```
[alejandros-mbp:PrinciplesOfBigDataProcessing alejandro$ scp -i CSCI-E88.pem -r ~/eclipse-workspace/csci-e-88-hw-1 centos@ec2-18-224-137-188.us-east-2.compute.amazonaws]
.com:~/
/etc/profile.d/lang.sh: line 19: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
FileGenerator.class                                                                                        100% 2351    20.4KB/s   00:00
Main.class                                                                                                 100%  817     6.9KB/s   00:00
.classpath                                                                                                 100%  295     2.6KB/s   00:00
org.eclipse.jdt.core.prefs                                                                                 100%  587     5.2KB/s   00:00
.project                                                                                                   100%  373     1.6KB/s   00:00
Main.java                                                                                                  100%  349     3.1KB/s   00:00
FileGenerator.java                                                                                         100% 1152    10.2KB/s   00:00
alejandros-mbp:PrinciplesOfBigDataProcessing alejandro$
```

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



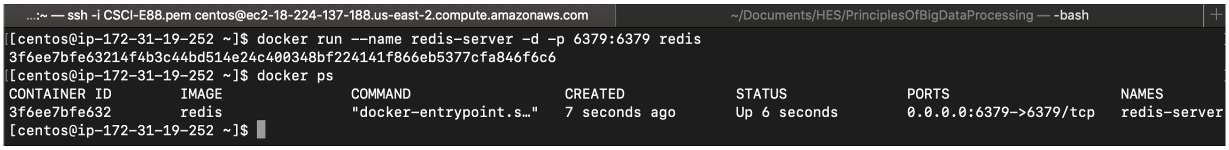
...ec2-18-224-137-188.us-east-2.compute.amazonaws.com
...cuments/HES/PrinciplesOfBigDataPr

```
[[centos@ip-172-31-19-252 bin]$ java edu.harvard.fall2019.cscie88.acg.hw1.Main 4 80
[[centos@ip-172-31-19-252 bin]$ ls -l
total 16
-rw-rw-r--. 1 centos centos 500 Sep  7 18:19 Alejandro_Chaparro_0.txt
-rw-rw-r--. 1 centos centos 496 Sep  7 18:19 Alejandro_Chaparro_1.txt
-rw-rw-r--. 1 centos centos 501 Sep  7 18:19 Alejandro_Chaparro_2.txt
-rw-rw-r--. 1 centos centos 502 Sep  7 18:19 Alejandro_Chaparro_3.txt
drwxr-xr-x. 3 centos centos  20 Sep  4 20:18 edu
[centos@ip-172-31-19-252 bin]$
```

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

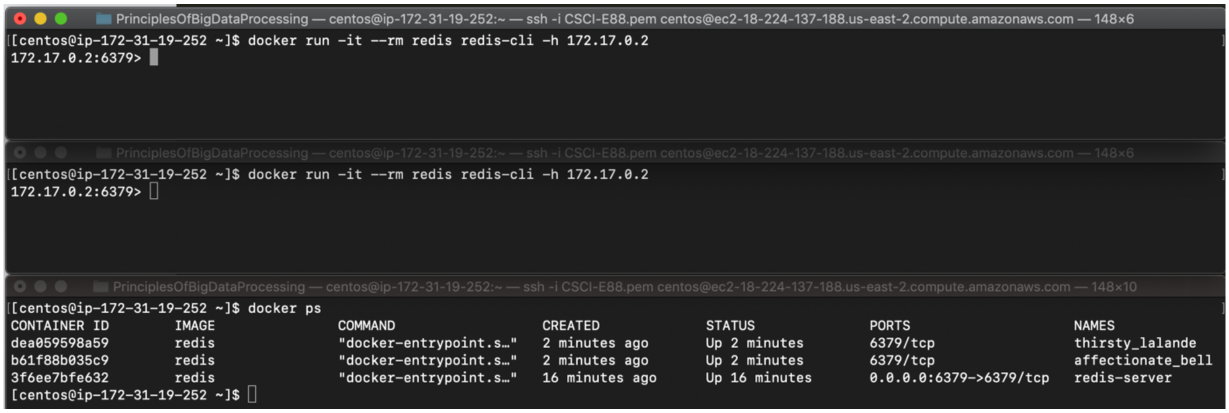
Show all the commands you used, in sequence, to start your Redis server and clients [15 points]

Screenshot:



...:~ — ssh -i CSCI-E88.pem centos@ec2-18-224-137-188.us-east-2.compute.amazonaws.com
~/Documents/HES/PrinciplesOfBigDataProcessing — -bash

```
[centos@ip-172-31-19-252 ~]$ docker run --name redis-server -d -p 6379:6379 redis
3f6ee7bfe63214f4b3c44bd514e24c400348bf224141f866eb5377cfa846f6c6
[centos@ip-172-31-19-252 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
3f6ee7bfe632        redis               "docker-entrypoint.s…"   7 seconds ago       Up 6 seconds        0.0.0.0:6379->6379/tcp   redis-server
[centos@ip-172-31-19-252 ~]$
```



PrinciplesOfBigDataProcessing — centos@ip-172-31-19-252:~ — ssh -i CSCI-E88.pem centos@ec2-18-224-137-188.us-east-2.compute.amazonaws.com — 148×6

```
[centos@ip-172-31-19-252 ~]$ docker run -it --rm redis redis-cli -h 172.17.0.2
172.17.0.2:6379>
```

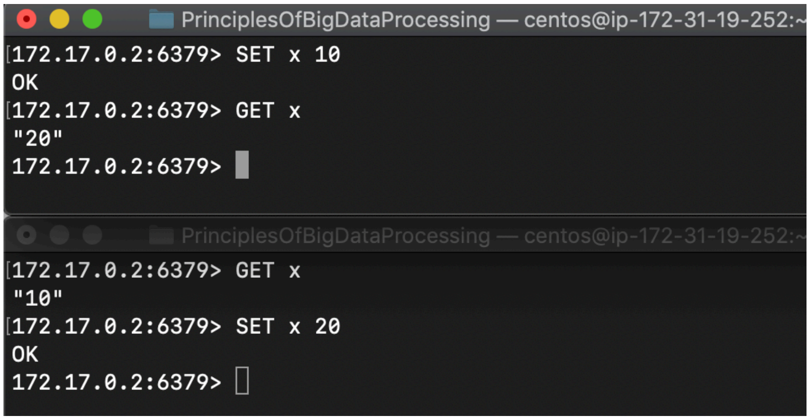
PrinciplesOfBigDataProcessing — centos@ip-172-31-19-252:~ — ssh -i CSCI-E88.pem centos@ec2-18-224-137-188.us-east-2.compute.amazonaws.com — 148×6

```
[centos@ip-172-31-19-252 ~]$ docker run -it --rm redis redis-cli -h 172.17.0.2
172.17.0.2:6379>
```

PrinciplesOfBigDataProcessing — centos@ip-172-31-19-252:~ — ssh -i CSCI-E88.pem centos@ec2-18-224-137-188.us-east-2.compute.amazonaws.com — 148×10

```
[centos@ip-172-31-19-252 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
dea059598a59        redis               "docker-entrypoint.s…"   2 minutes ago       Up 2 minutes        6379/tcp                 thirsty_lalande
b61f88b035c9        redis               "docker-entrypoint.s…"   2 minutes ago       Up 2 minutes        6379/tcp                 affectionate_bell
3f6ee7bfe632        redis               "docker-entrypoint.s…"   16 minutes ago      Up 16 minutes       0.0.0.0:6379->6379/tcp   redis-server
[centos@ip-172-31-19-252 ~]$
```

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



PrinciplesOfBigDataProcessing — centos@ip-172-31-19-252:~

```
[172.17.0.2:6379> SET x 10
OK
[172.17.0.2:6379> GET x
"20"
172.17.0.2:6379>
```

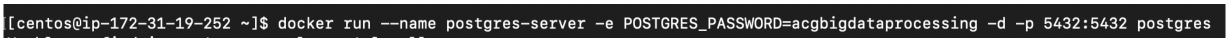
PrinciplesOfBigDataProcessing — centos@ip-172-31-19-252:~

```
[172.17.0.2:6379> GET x
"10"
[172.17.0.2:6379> SET x 20
OK
172.17.0.2:6379>
```

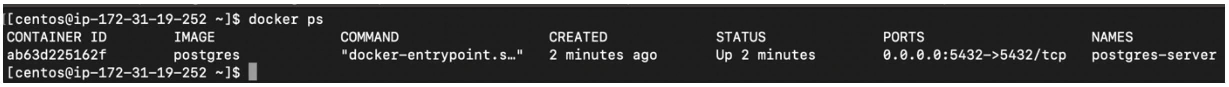
Problem 4: [25 points] Run Postgres DB as Docker container and demonstrate that it works

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

Screenshot:

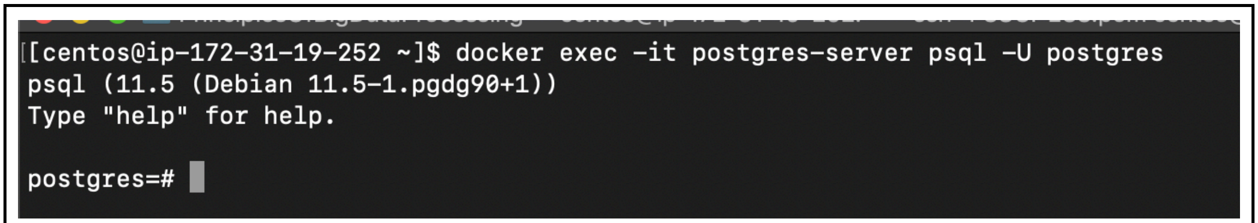


```
[centos@ip-172-31-19-252 ~]$ docker run --name postgres-server -e POSTGRES_PASSWORD=acgbigdataprocessing -d -p 5432:5432 postgres
```



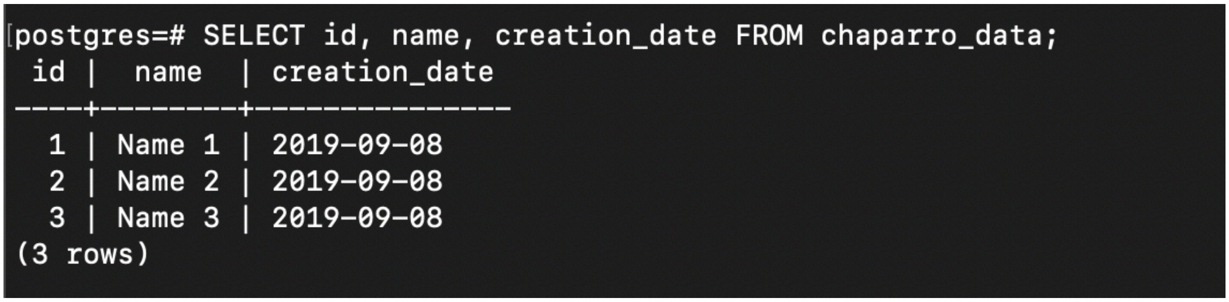
```
[centos@ip-172-31-19-252 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
ab63d225162f        postgres            "docker-entrypoint.s…"   2 minutes ago       Up 2 minutes        0.0.0.0:5432->5432/tcp   postgres-server
[centos@ip-172-31-19-252 ~]$
```

Show how you connect to the DB [5 points]



```
[[centos@ip-172-31-19-252 ~]$ docker exec -it postgres-server psql -U postgres
psql (11.5 (Debian 11.5-1.pgdg90+1))
Type "help" for help.

postgres=#
```

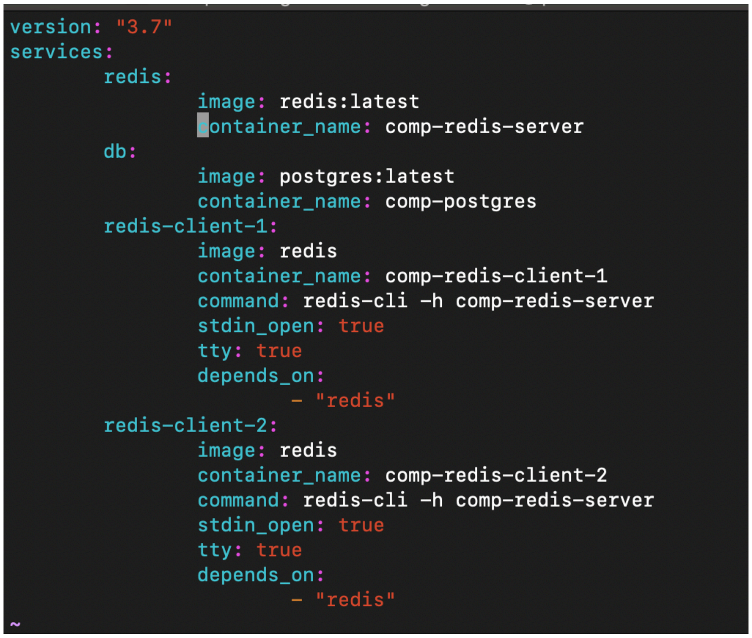
Show results of querying your database for all records. [10 points]



```
[postgres=# SELECT id, name, creation_date FROM chaparro_data;
 id |  name  | creation_date
----+--------+---------------
  1 | Name 1 | 2019-09-08
  2 | Name 2 | 2019-09-08
  3 | Name 3 | 2019-09-08
(3 rows)
```

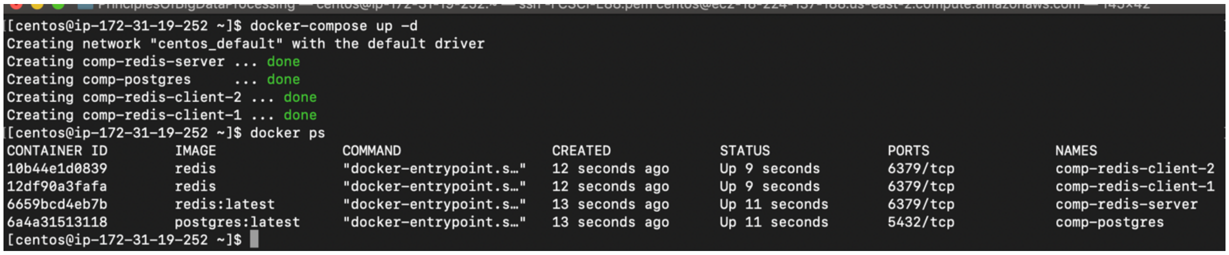
Problem 5: [Bonus, 15 points]: Start multiple Docker container via Compose

Show your Docker Compose configuration [7 points]

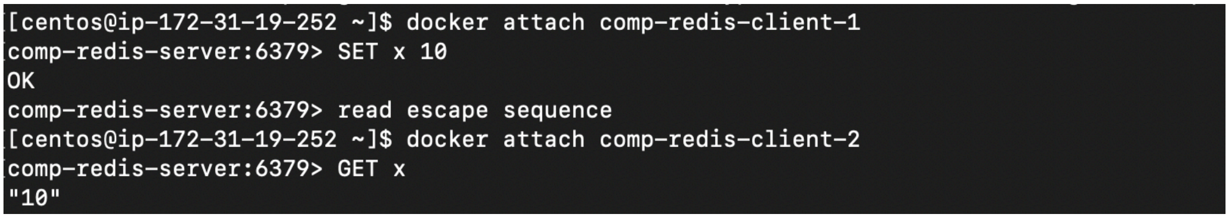


```
version: "3.7"
services:
        redis:
                image: redis:latest
                container_name: comp-redis-server
        db:
                image: postgres:latest
                container_name: comp-postgres
        redis-client-1:
                image: redis
                container_name: comp-redis-client-1
                command: redis-cli -h comp-redis-server
                stdin_open: true
                tty: true
                depends_on:
                        - "redis"
        redis-client-2:
                image: redis
                container_name: comp-redis-client-2
                command: redis-cli -h comp-redis-server
                stdin_open: true
                tty: true
                depends_on:
                        - "redis"
~
```

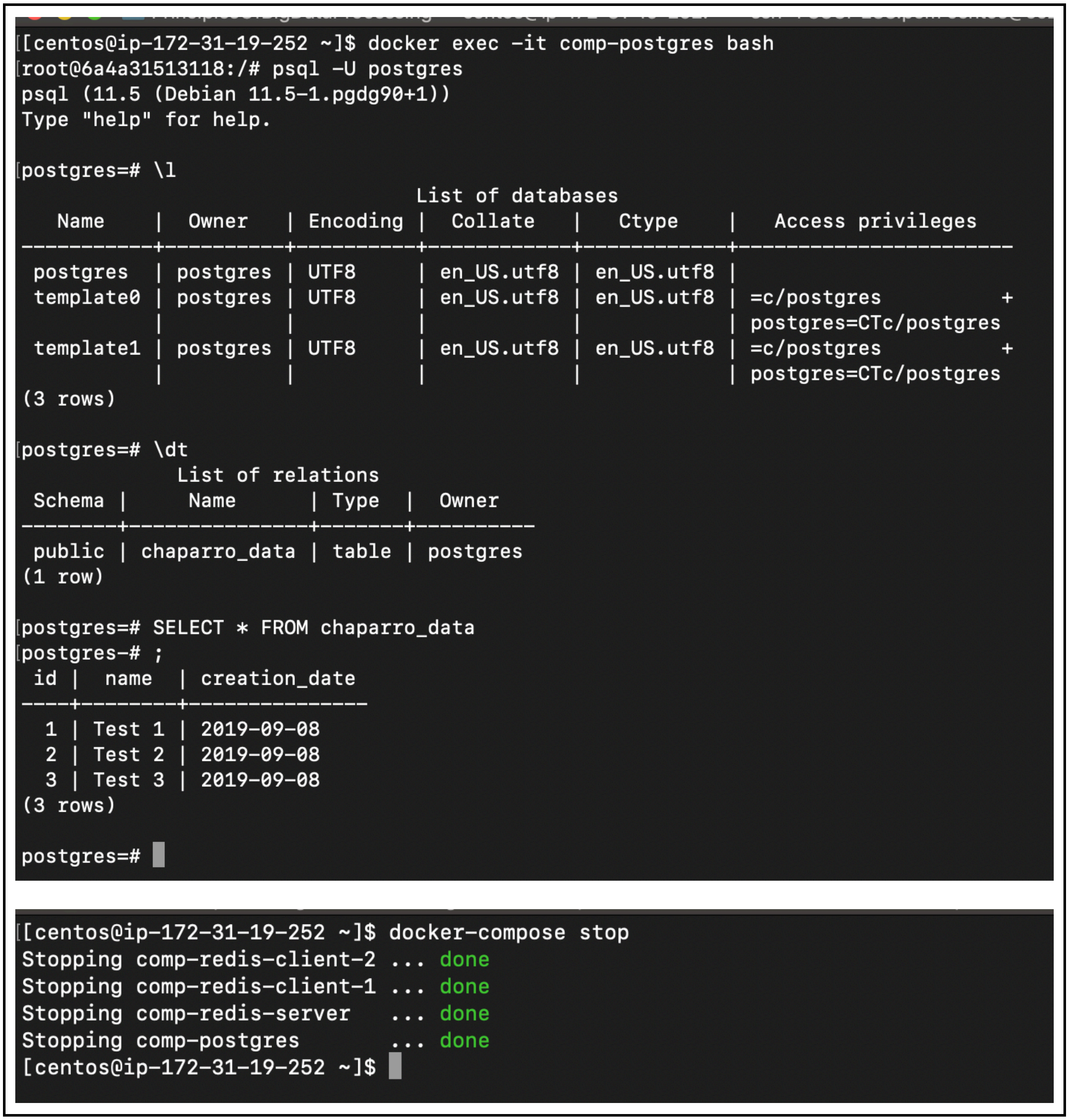
Show that the Redis server, 2 Redis clients, Postgres server are all functional [8 points]



```
[centos@ip-172-31-19-252 ~]$ docker-compose up -d
Creating network "centos_default" with the default driver
Creating comp-redis-server ... done
Creating comp-postgres     ... done
Creating comp-redis-client-2 ... done
Creating comp-redis-client-1 ... done
[centos@ip-172-31-19-252 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
10b44e1d0839        redis               "docker-entrypoint.s…"   12 seconds ago      Up 9 seconds        6379/tcp            comp-redis-client-2
12df90a3fafa        redis               "docker-entrypoint.s…"   12 seconds ago      Up 9 seconds        6379/tcp            comp-redis-client-1
6659bcd4eb7b        redis:latest        "docker-entrypoint.s…"   13 seconds ago      Up 11 seconds       6379/tcp            comp-redis-server
6a4a31513118        postgres:latest     "docker-entrypoint.s…"   13 seconds ago      Up 11 seconds       5432/tcp            comp-postgres
[centos@ip-172-31-19-252 ~]$ 
```



```
[centos@ip-172-31-19-252 ~]$ docker attach comp-redis-client-1
comp-redis-server:6379> SET x 10
OK
comp-redis-server:6379> read escape sequence
[centos@ip-172-31-19-252 ~]$ docker attach comp-redis-client-2
comp-redis-server:6379> GET x
"10"
```



```
[[centos@ip-172-31-19-252 ~]$ docker exec -it comp-postgres bash
[root@6a4a31513118:/# psql -U postgres
psql (11.5 (Debian 11.5-1.pgdg90+1))
Type "help" for help.

[postgres=# \l
                                 List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------+----------+----------+------------+------------+-----------------------
 postgres  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
(3 rows)

[postgres=# \dt
             List of relations
 Schema |     Name      | Type  |  Owner
--------+---------------+-------+----------
 public | chaparro_data | table | postgres
(1 row)

[postgres=# SELECT * FROM chaparro_data
[postgres-# ;
 id |  name  | creation_date
----+--------+---------------
  1 | Test 1 | 2019-09-08
  2 | Test 2 | 2019-09-08
  3 | Test 3 | 2019-09-08
(3 rows)

postgres=# 
```

```
[[centos@ip-172-31-19-252 ~]$ docker-compose stop
Stopping comp-redis-client-2 ... done
Stopping comp-redis-client-1 ... done
Stopping comp-redis-server   ... done
Stopping comp-postgres       ... done
[centos@ip-172-31-19-252 ~]$ 
```