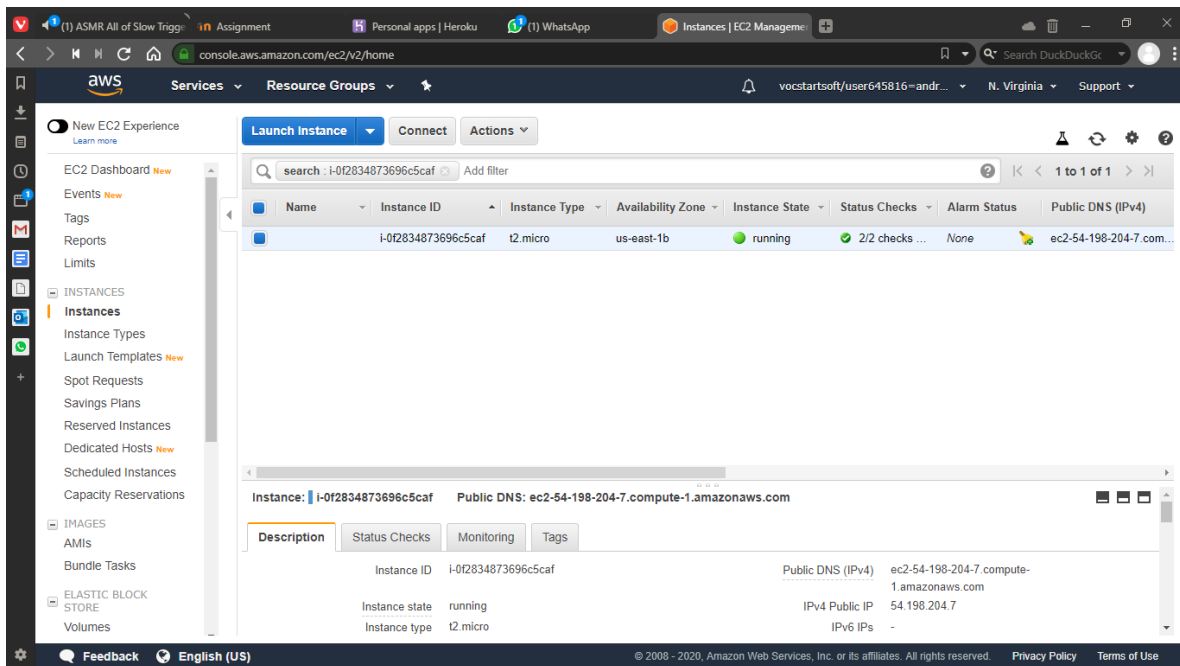
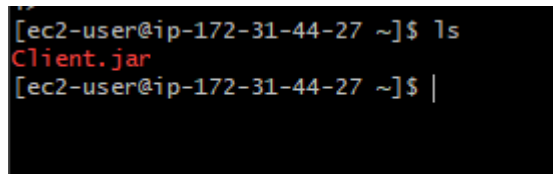


Andres Villamil
2130104

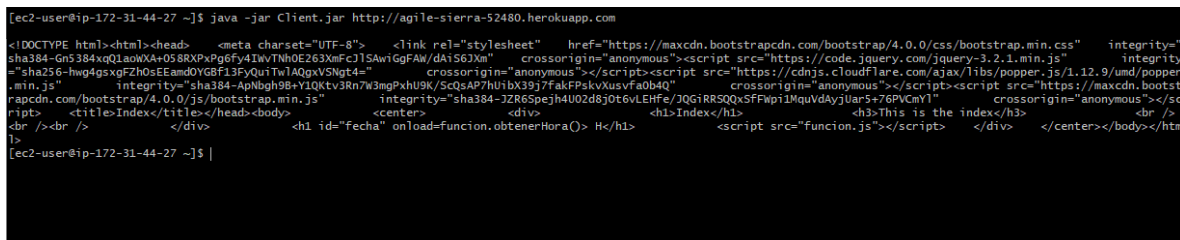
1. Crear una máquina virtual de Linux en AWS.



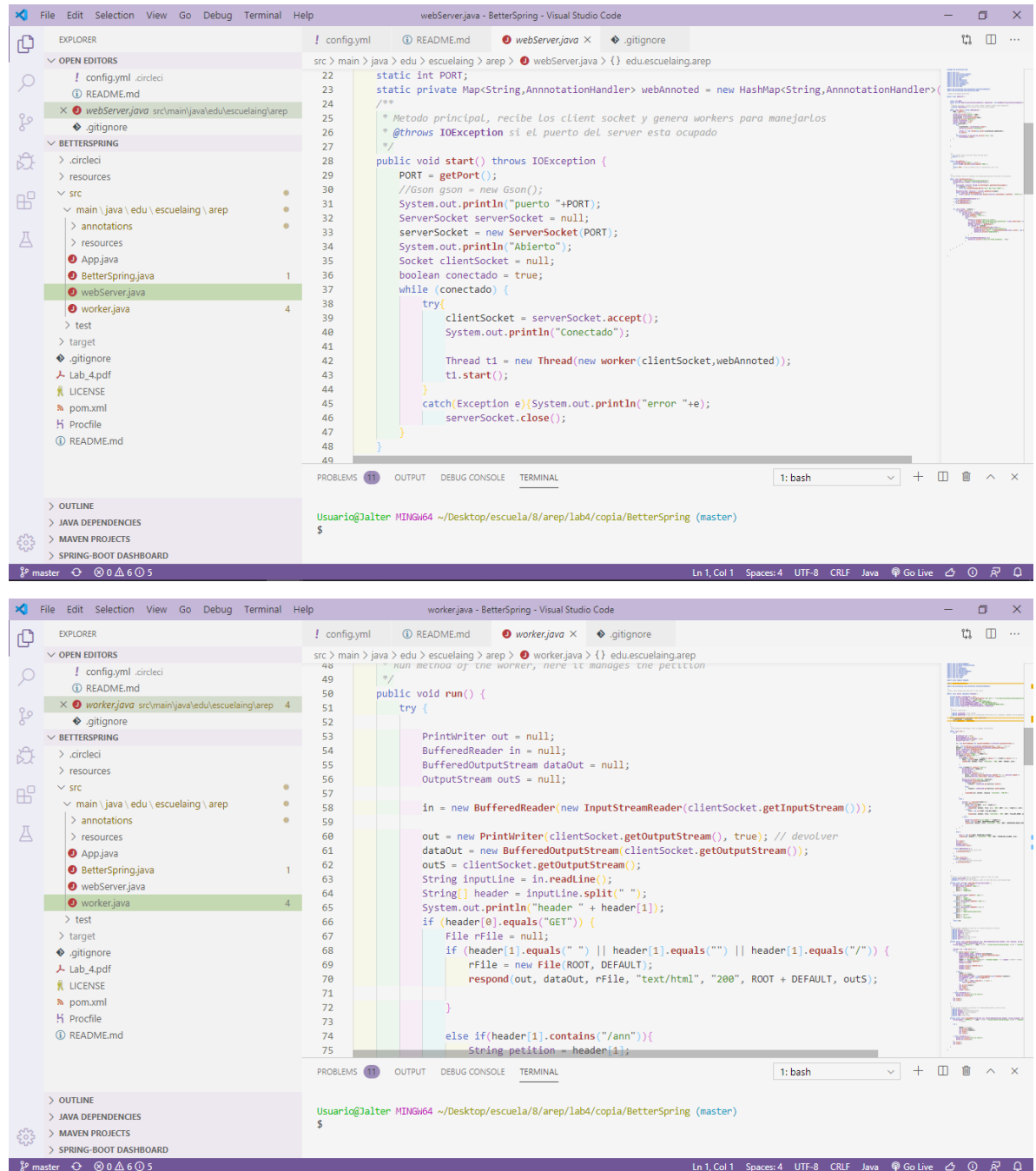
2. Desplegar un cliente que lea URLs e imprima contenidos en consola, en la máquina virtual de AWS.



3. Conectar el cliente desplegado a una aplicación Web desplegada en Heroku. usando el servidor Web que desarrolló



4. Modifique su servidor Web para que responda múltiples peticiones de manera concurrente.



5. Modifique su cliente para probar su servidor web haciendo muchas peticiones en paralelo.

Andres Villamil
2130104

```
Client.java X Client.class Worker.java
Client.java > Client > main(String[])
14 /**
15  * Metodo principal, recibe los client socket y genera workers para manejarlos
16  *
17  * @throws IOException si el puerto del server esta ocupado
18  */
19 public static void main(String[] args) throws IOException {
20
21     ExecutorService executioner = Executors.newFixedThreadPool(Threads);
22     ArrayList<URL> urls = new ArrayList<URL>();
23     URL a = new URL("https://agile-sierra-52480.herokuapp.com");
24     URL b = new URL("http://agile-sierra-52480.herokuapp.com/ann/hola?andres");
25     URL c = new URL("http://agile-sierra-52480.herokuapp.com/ann/hello");
26     URL d = new URL("http://agile-sierra-52480.herokuapp.com/img1.jpg");
27     URL e = new URL("http://agile-sierra-52480.herokuapp.com/img2.png");
28     urls.add(e);
29     urls.add(a);
30     urls.add(b);
31     urls.add(c);
32     urls.add(d);
33     urls.add(e);
34     int i = 0;
35     while (i < urls.size()) {
36
37         System.out.println("Conectado");
38         executioner.execute(new Thread(new Worker(urls.get(i))));
39         i++;
40
41     }
42 }
43 }
44
45 }
```

```
1 /**
2  * This class manage each petition to the server
3  */
4
5 public class Worker implements Runnable {
6
7     URL url;
8
9     public Worker(URL url) {
10         this.url = url;
11     }
12
13     /**
14      * Run method of the worker, here it manages the petition
15      */
16     public void run() {
17         try {
18
19             BufferedReader reader = new BufferedReader(new InputStreamReader(url.openStream()));
20             String inputLine = null;
21             while ((inputLine = reader.readLine()) != null) {
22                 System.out.println(inputLine);
23             }
24         } catch (IOException x) {
25             System.err.println(x);
26         }
27     }
28 }
29
30 }
```

Andres Villamil
2130104

6. Mida el desempeño de su servidor web variando el número de hilos que soporta y aumentando o disminuyendo el número de peticiones concurrentes.
7. Pruebe tanto contenidos estáticos como dinámicos.

[illegible]