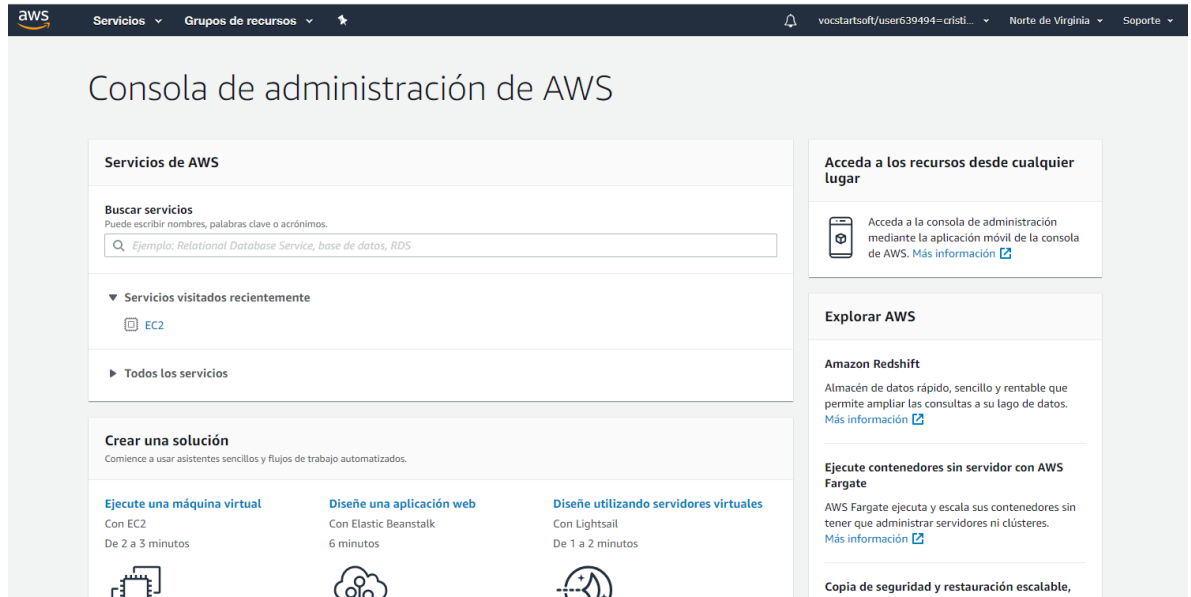
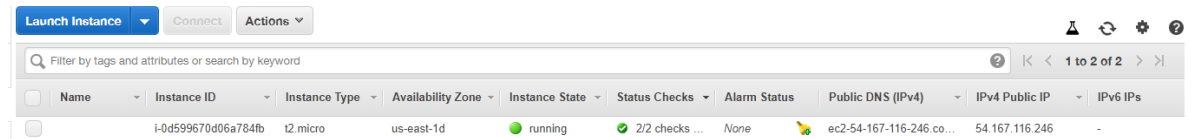


Cristian David López Arévalo

1. Acceda a la consola de administración de AWS



2. Cree una máquina virtual linux siguiendo los pasos en: <https://aws.amazon.com/es/getting-started/tutorials/launch-a-virtual-machine/>



3. Conéctese a la máquina virtual usando ssh. Verifique que está en la máquina virtual introduciendo comandos simples como: whoami, ls, pwd.

Ssh

```
N-AWS>ssh -i "intro-aws-key.pem" ec2-user@ec2-54-167-116-246.compute-1.amazonaws.com
```

Whoami

```
[ec2-user@ip-172-31-85-181 ~]$ whoami
ec2-user
```

Ls

```
[ec2-user@ip-172-31-85-181 ~]$ ls
arep
```

Pwd

```
[ec2-user@ip-172-31-85-181 ~]$ pwd  
/home/ec2-user
```

4. Verifique que java está instalado. Note que el compilador de java (javac) no está instalado en la máquina virtual.

```
[ec2-user@ip-172-31-85-181 ~]$ java -version  
openjdk version "1.8.0_242"  
OpenJDK Runtime Environment (build 1.8.0_242-b08)  
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```

5. Salga del ssh usando "exit".

```
[ec2-user@ip-172-31-85-181 ~]$ exit  
logout  
Connection to ec2-54-167-116-246.compute-1.amazonaws.com closed.
```

6. En su **máquina local**, usando netbeans cree un cliente que se pueda conectar a una url e imprimir la respuesta de esa url en pantalla. Observe que el código de ejemplo recibe la url como el primer argumento en la línea de comandos.

```

7 public class ClientServer implements Runnable {
8
9     private URL url;
10    private static int threads;
11    static AtomicInteger succesA = new AtomicInteger(0);
12    static AtomicInteger failA = new AtomicInteger(0);
13
14    public ClientServer(String url) {
15        try {
16            this.url = new URL(url);
17        } catch (MalformedURLException e) {}
18        System.out.println("MalformedURLException: " + e);
19        e.printStackTrace();
20    }
21
22    Run | Debug
23    public static void main(final String[] args) {
24        ClientServer cs = new ClientServer(args[0]);
25        threads = Integer.parseInt(args[1]);
26        Thread[] peticiones = new Thread[threads];
27        for (int i = 0; i < threads; i++) {
28            peticiones[i] = new Thread(cs);
29            peticiones[i].start();
30        }
31        for (int i = 0; i < threads; i++) {
32            try {
33                peticiones[i].join();
34            } catch (InterruptedException e) {
35                e.printStackTrace();
36                System.out.println("InterruptedException: " + e);
37            }
38        }
39        System.out.println("succes: " + succesA + " " + " fail : " + failA);
40    }
41
42    @Override
43    public void run() {
44        try {
45            BufferedReader reader = new BufferedReader(new InputStreamReader(url.openStream()));
46            String inputline = null;
47            while ((inputline = reader.readLine()) != null) {
48                // System.out.println(inputline);
49            }
50            succesA.addAndGet(1);
51        } catch (final IOException x) {
52            System.err.println(x);
53            failA.addAndGet(1);
54        }
55    }
56 }
57

```

7. Pruebe su cliente en la máquina local con un comando similar al siguiente:

```
>mvn exec:java -Dexec.mainClass="edu.escuelaing.arep.cliente.ClientServer" -Dexec.args="http://www.google.com 1"
[INFO] Scanning for projects...
[INFO] -----< edu.escuelaing.arep:cliente >-----
[INFO] Building cliente 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ cliente ---
succes: 1 fail : 0
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.991 s
[INFO] Finished at: 2020-03-09T21:35:34-05:00
[INFO] -----
```

8. Suba el proyecto compilado a su máquina virtual usando sftp.

```
N-AWS>sftp -i "intro-aws-key.pem" ec2-user@ec2-54-167-116-246.compute-1.amazonaws.com
Connected to ec2-user@ec2-54-167-116-246.compute-1.amazonaws.com.
sftp> put -r target/classes
```

9. Ejecute el cliente que instaló en su máquina virtual de AWS para conectarse a la aplicación que instaló en Heroku durante el parcial o el taller.

```
[ec2-user@ip-172-31-85-181 target]$ java edu/escuelaing/arep/cliente/ClientServer https://shielded-coast-58752.herokuapp.com/index.html 400
```

```
[ec2-user@ip-172-31-85-181 target]$ java edu/escuelaing/arep/cliente/ClientServer https://shielded-coast-58752.herokuapp.com/index.html 400
succes: 400 fail : 0
```

10. Borre las instancias y unidades de almacenamiento en su cuenta AWS para no generar costos.

The screenshot shows the AWS Management Console interface for EC2 instances. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below these is a search bar labeled 'Filter by tags and attributes or search by keyword'. A table lists EC2 instances with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, and Status Checks. Two instances are visible: one with ID i-0d599670d06a784fb in a 'running' state, and another with ID i-00e00d1271cbe1b in a 'stopped' state. A context menu is open over the first instance, showing various actions. The 'Instance State' option is highlighted, and its sub-menu is also open, displaying options: Start, Stop, Stop - Hibernate, Reboot, and Terminate. Below the table, the 'Description' tab is selected for the instance i-0d599670d06a784fb, showing details like Instance ID and Instance State.