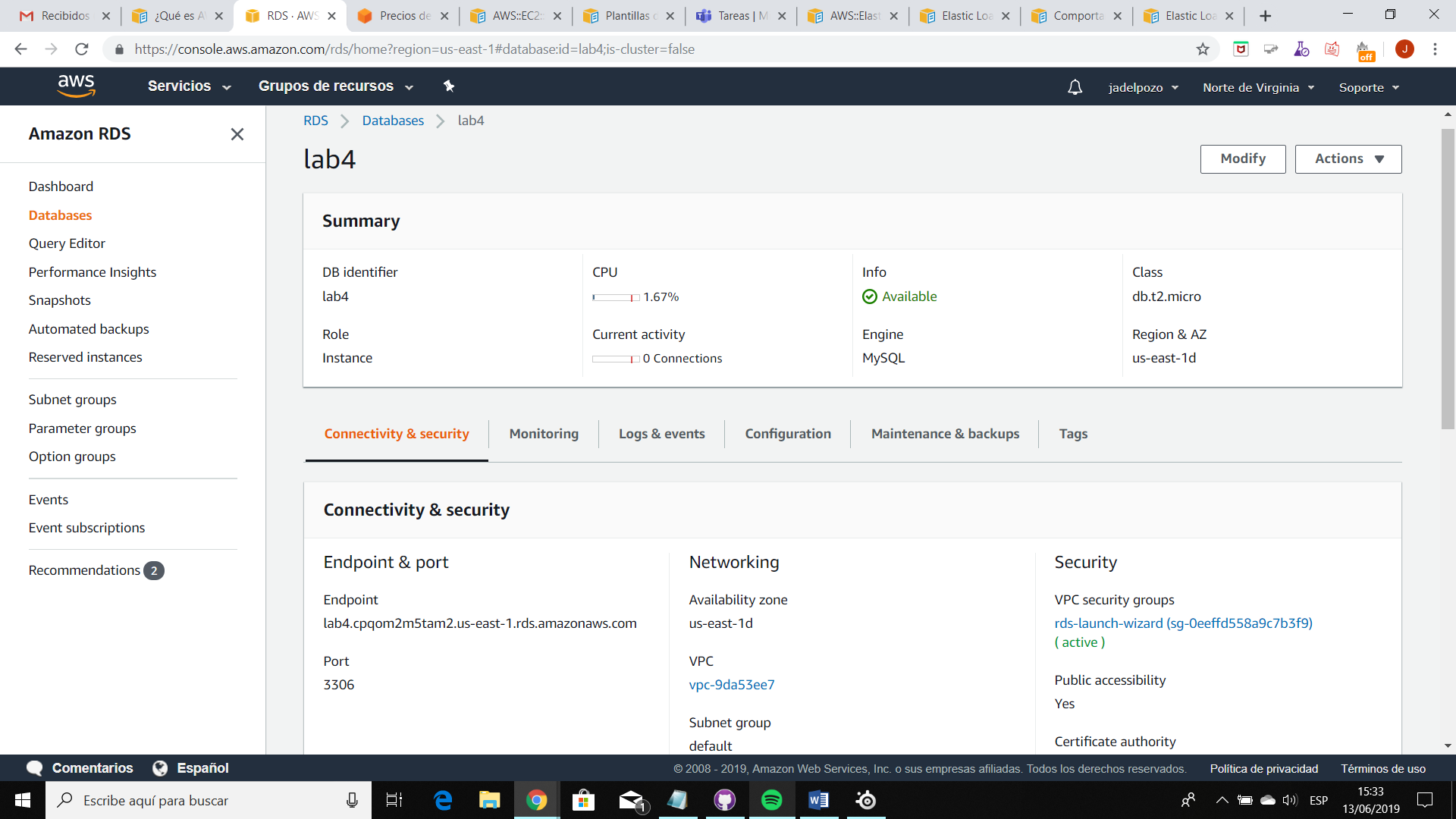
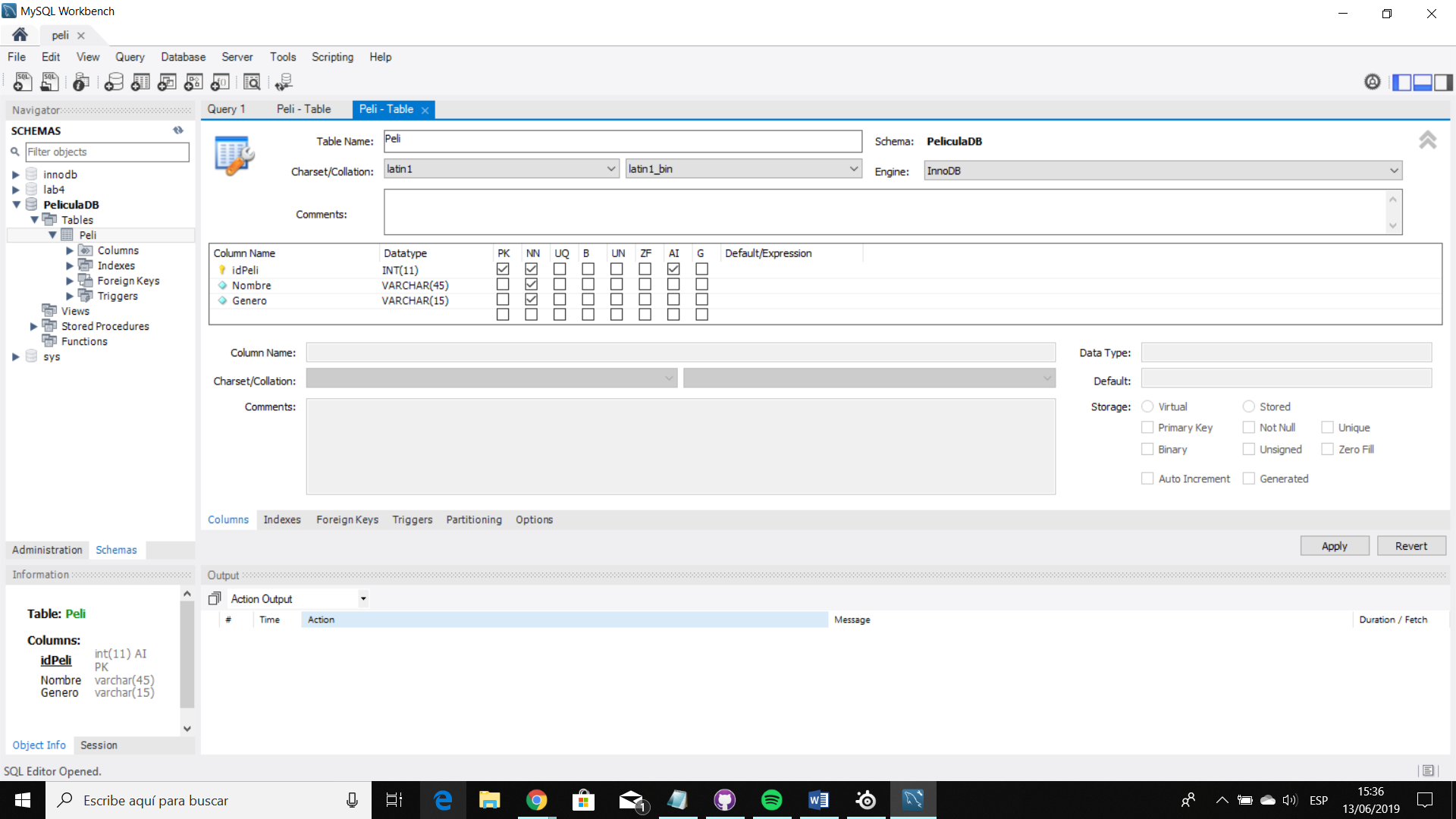
Pasos para la elaboración:

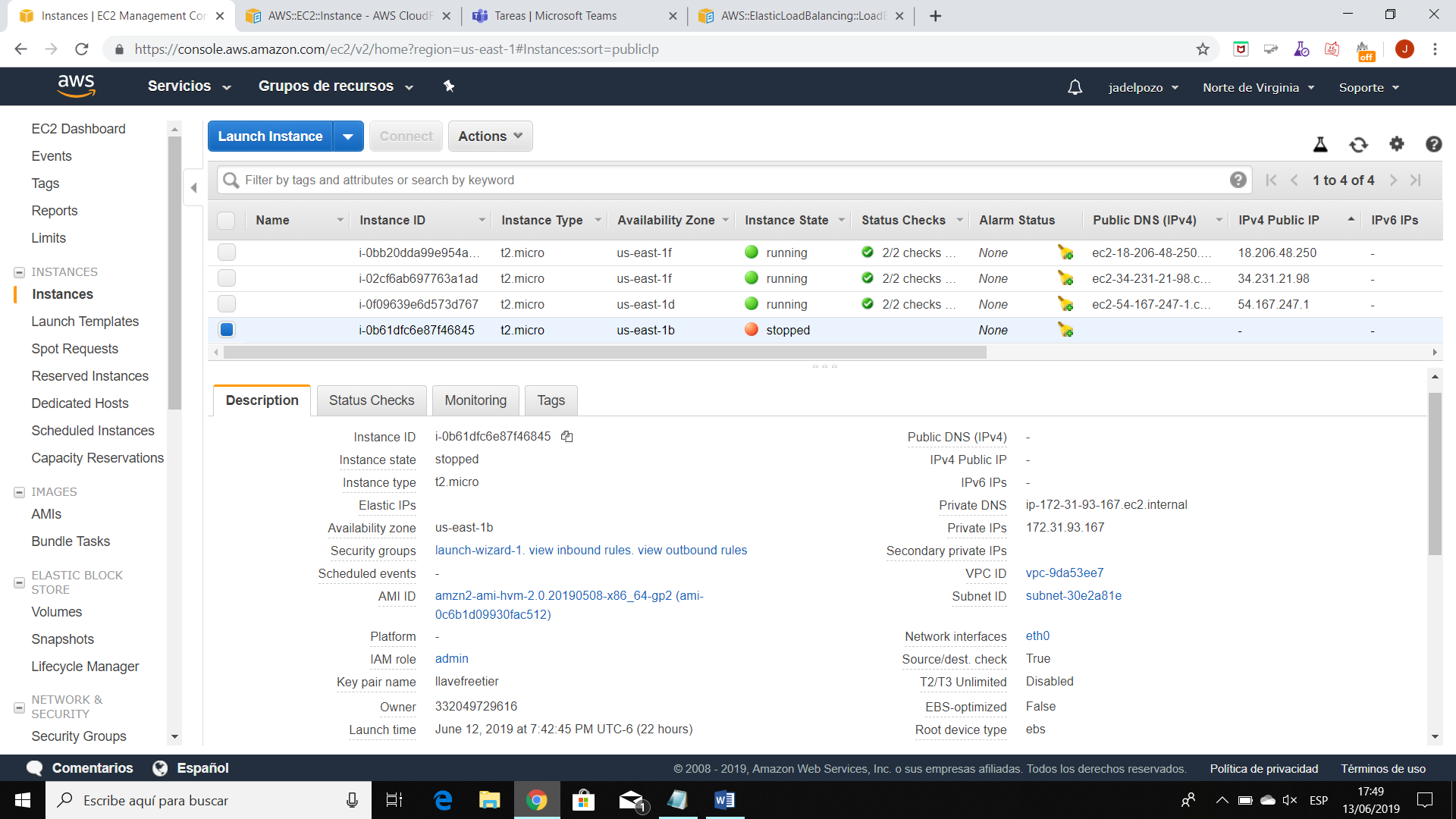
1. Primero se hizo una investigación sobre cómo realizar un CRUD y crear una base de datos en AWS.
2. Se creo una base de datos MySQL usando el servicio RDS de AWS.



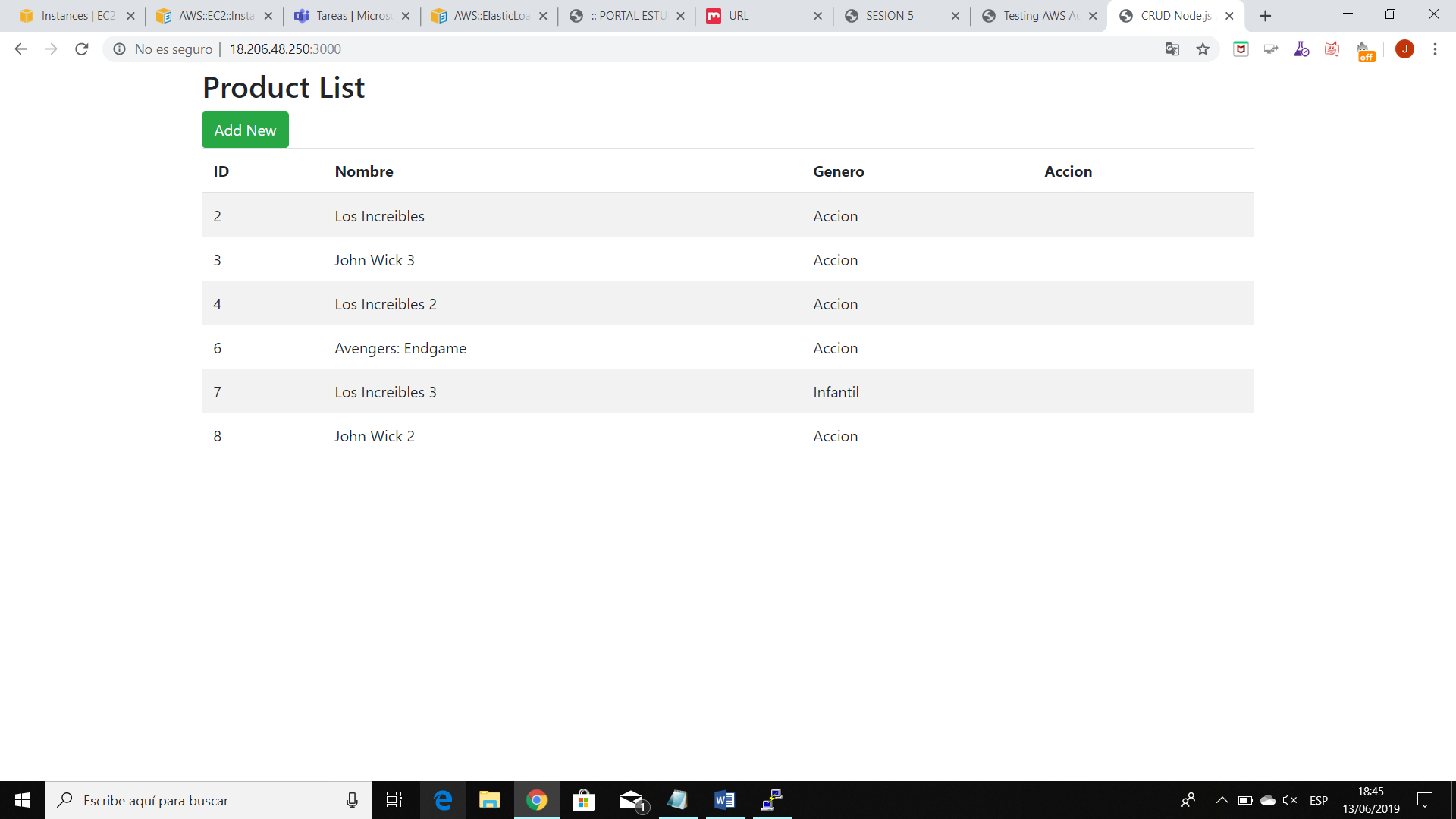
1. Se hicieron pruebas de conexión desde MySQL Workbench y una vez conectado se creo un Schema “PeliculaDB” y una tabla de nombre “Peli” con las columnas idPeli, Nombre y Genero.



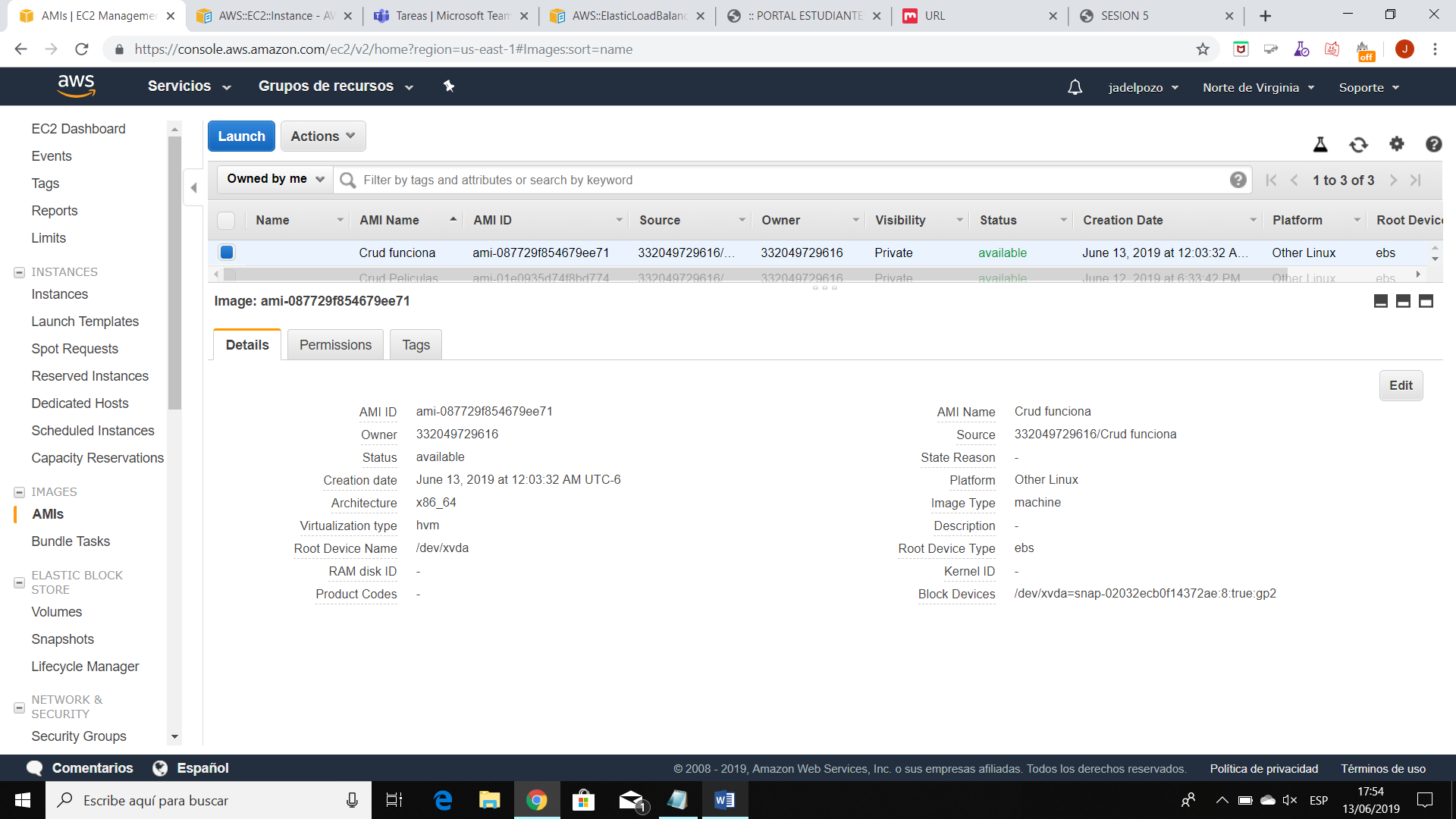
1. Se realizo el CRUD en Node js y se hicieron pruebas locales. Una vez que ya funciono local se creo una instancia para que funcionara como servidor web y ahí se subió la aplicación por medio de Git.



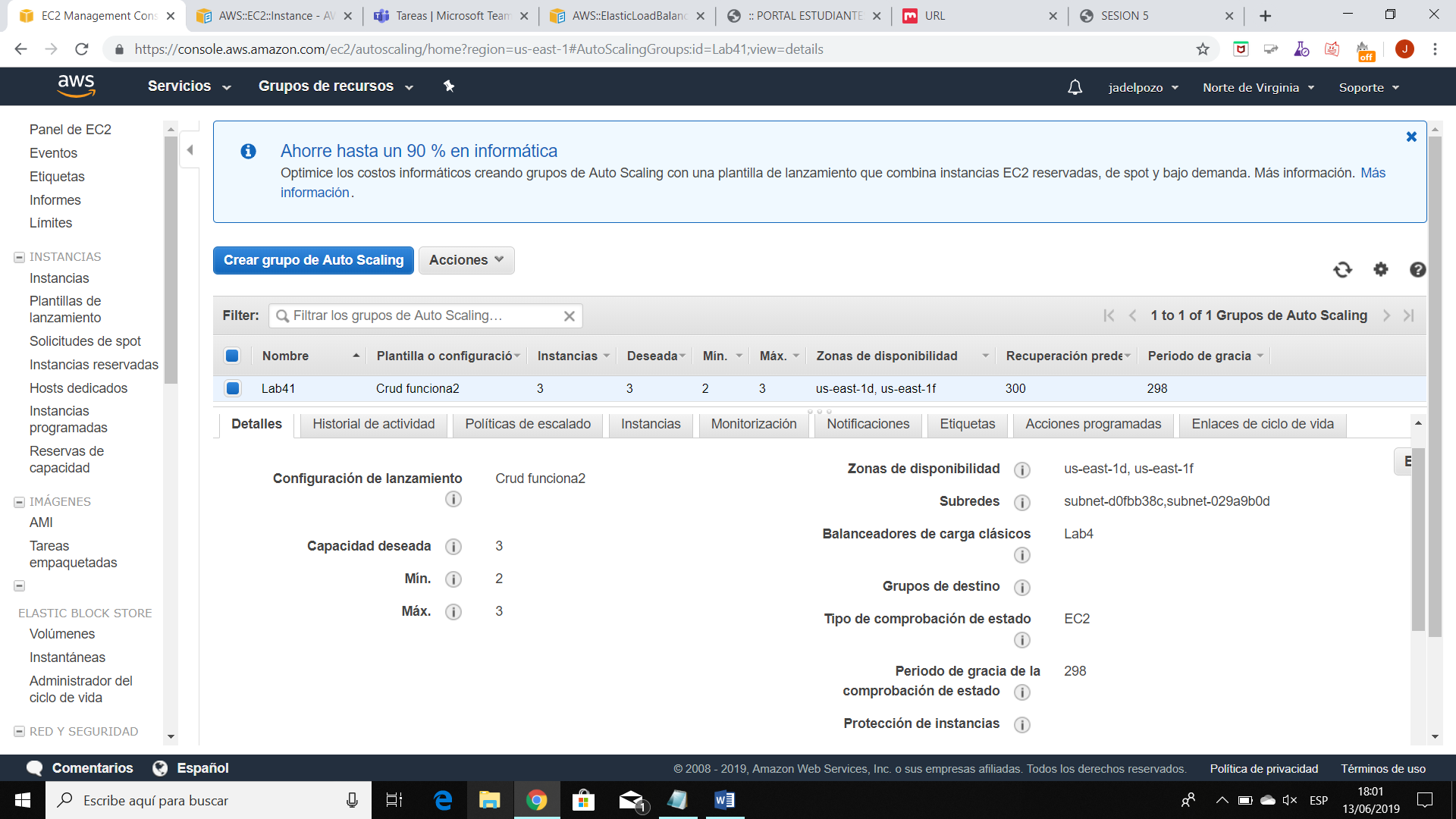
La aplicación corriendo desde la ip publica del servidor web.



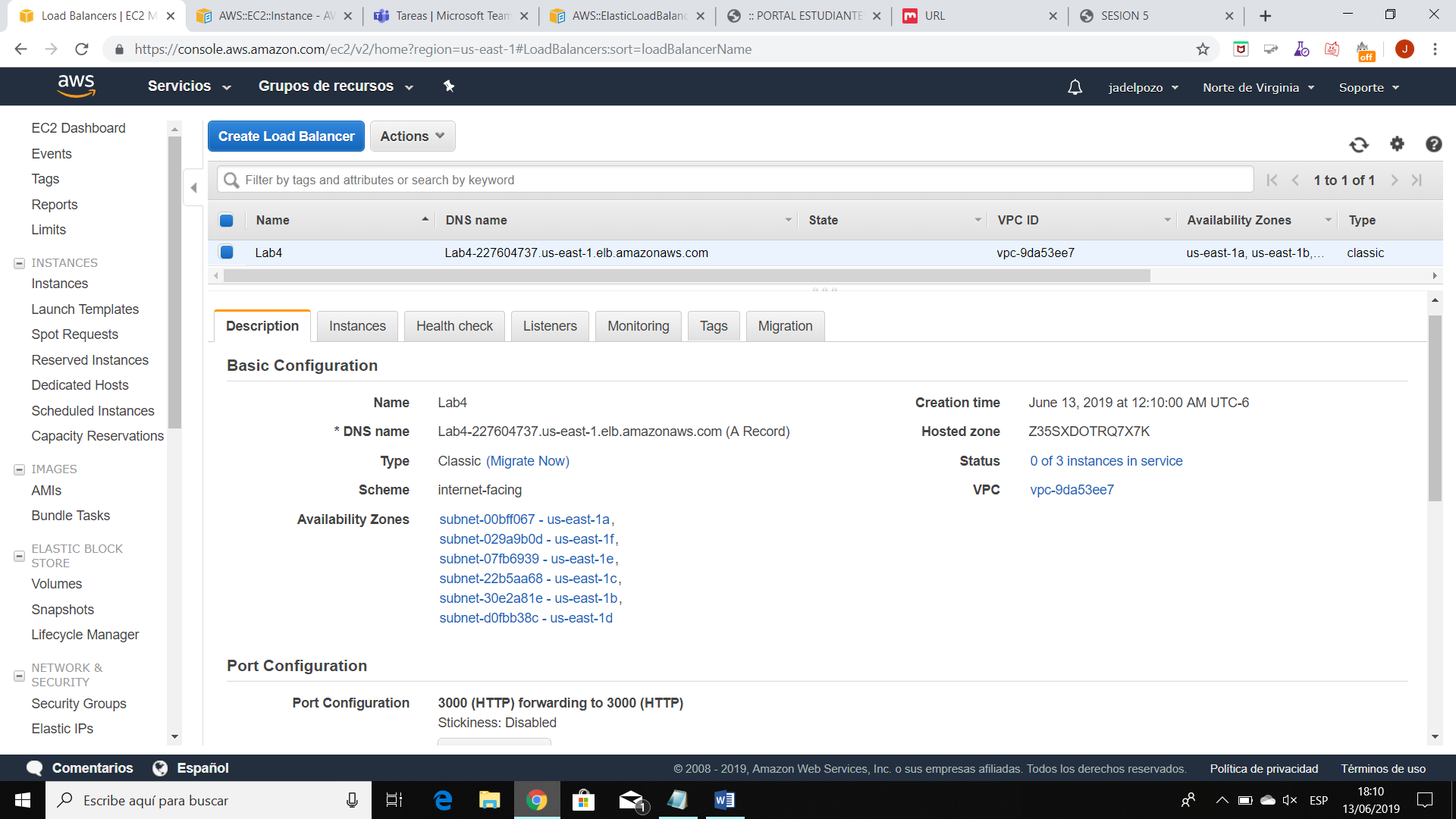
1. Se creo una imagen de la maquina para poder usarla en el auto scalling group.



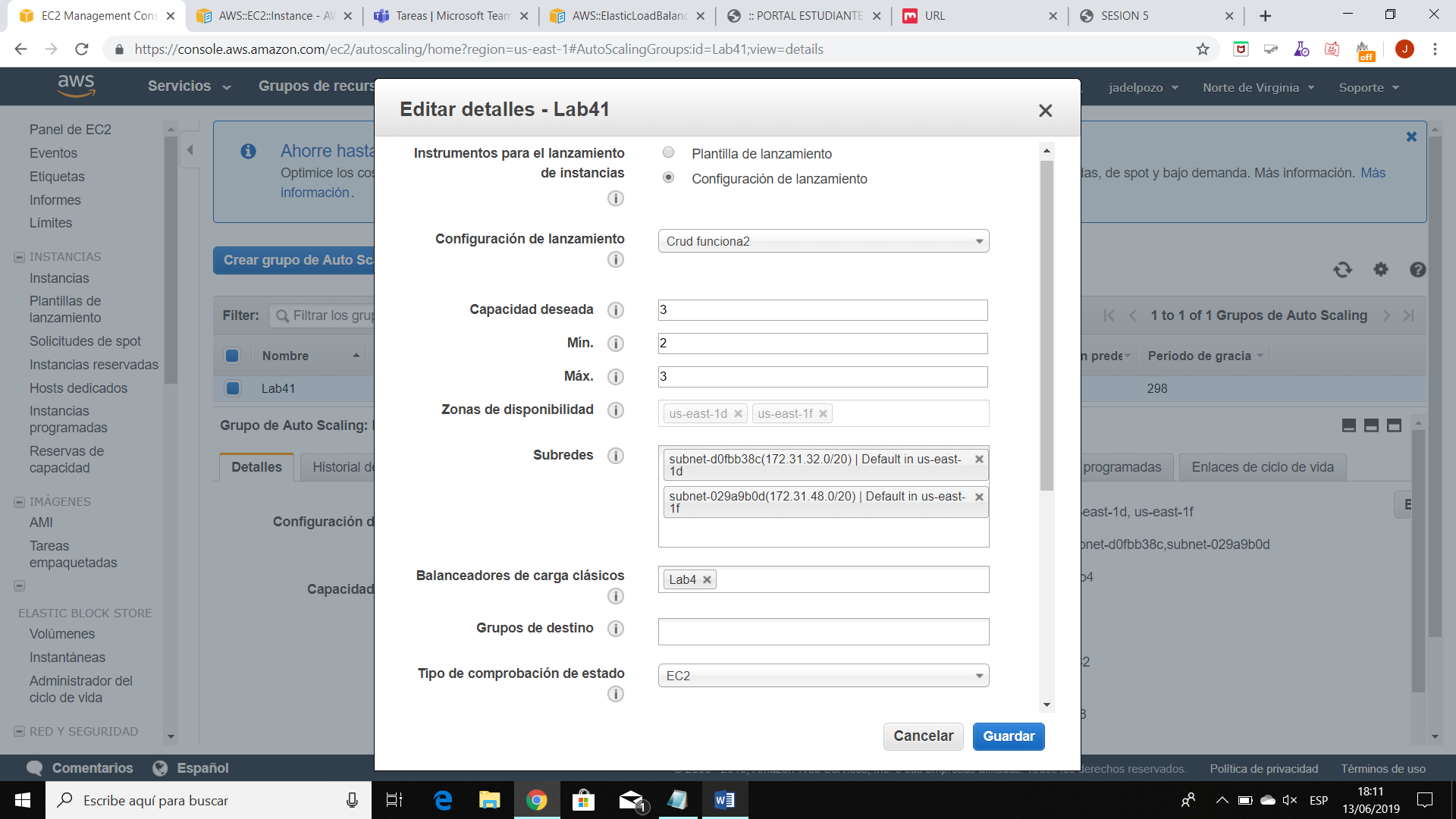
1. Se creo el auto scalling group utilizando la imagen creada previamente para usarla como parte del Launch Configuration y se configuro una arquitectura de 2 servidores web en diferentes regiones y que pueda crecer hasta 3 maquinas si se consume el 20% del CPU.



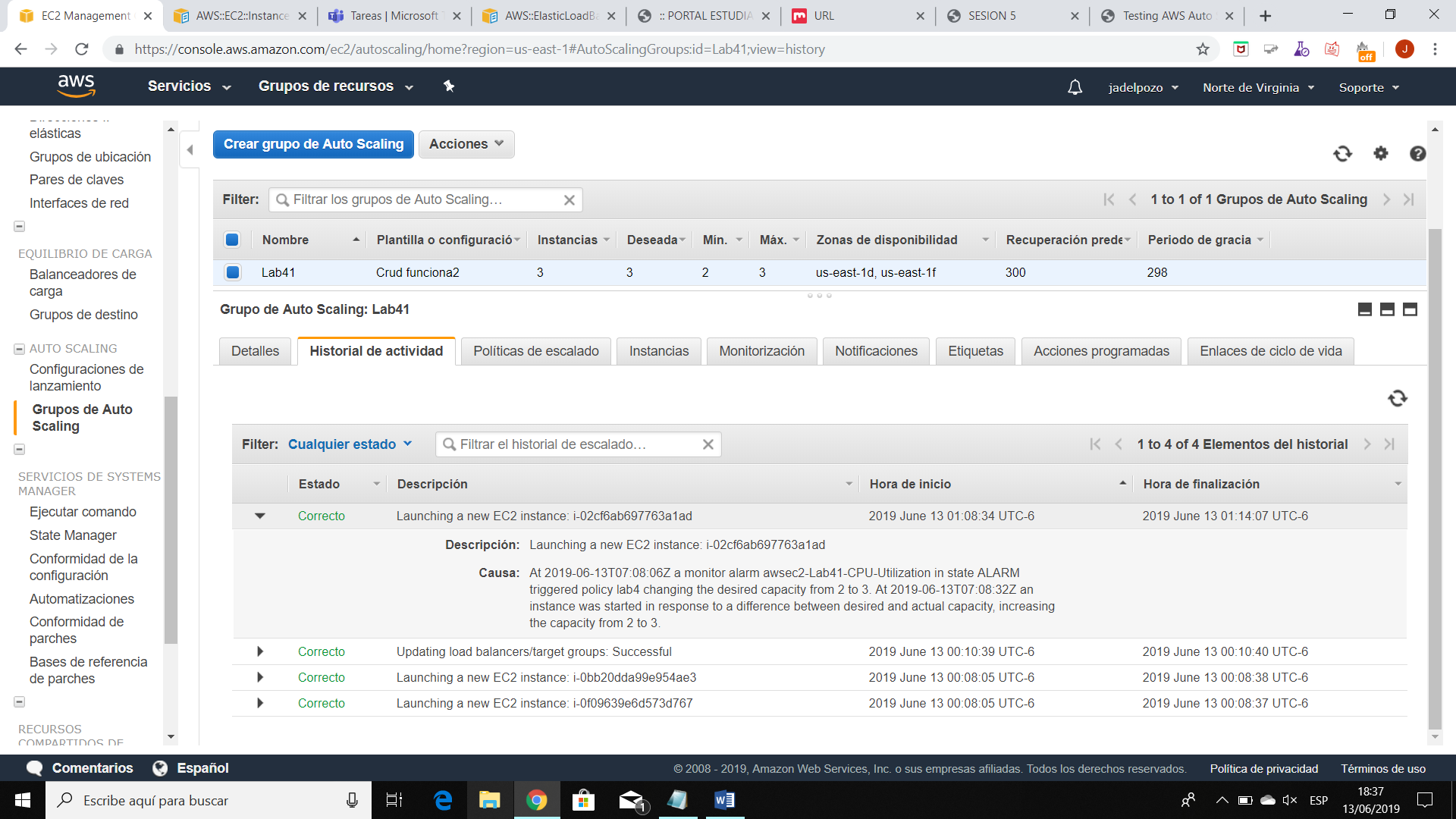
1. Se creo un Load Balancer para manejar el auto scalling group.



1. Luego se asigno el Auto Scalling Group al Load Balancer creado previamente.



1. Luego se hicieron pruebas de carga en los servidores web para probar el Auto Scalling utilizando el comando: [root@ip]# stress --cpu 1 --timeout 300 para forzar la computadora y así hacer que el Auto Scalling cree otro servidor web. En la imagen se puede observar como en el historial aparece que saltó la alarma para crear una nueva instancia llegando al máximo de instancias aceptada por el Scalling Group.



1. La aplicación web corriendo desde el DNS proporcionado por el Load Balancer.

