Aplicación distribuida segura en todos sus frentes

Luis Alejandro Jaramillo Rincon

01/Oct/2020

1 Introduction

El objetivo de este laboratorio es crear dos aplicaciones web seguras, para lo que vamos a crear pares de llaves, así estableceremos una conexión http segura (https), para ello debemos modificar la aplicación para que use los certificados creados. Después de resolver ese cero, procedemos a crear las imágenes de Docker de nuestras dos aplicaciones web, en la imagen incluiremos los certificados. Por último subiremos la aplicación a AWS.

2 Herramientas

- Java como lenguaje de programación
- Maven para manejar el cíclo vida del proyecto
- Spark Java. "A micro framework for creating web applications in Kotlin and Java 8 with minimal effort"
- Keytool de java para generar y administrar las llaves y los certificados

3 Arquitectura

Mediante una conexión https vamos a acceder por nuestro browser a nuestra primera aplicación web, la cual contará con pantalla de login, la cual nos dirigirá si nos autenticamos correctamente a una pantalla donde debemos ingresar unos números, a los cuales se les va a realizar un ordenamiento. Cuando ingresemos los números, nuestro servicio 1 se va a conectar a nuestro servicio 2 mediante https, en el servicio dos se va a hacer el ordenamiento va a retornarnos una pagina a la cual va a acceder el servicio 1.

Lo primero que se hizo fue crear las imágenes en Docker, donde se copiaron los certificados de los dos servicios, posteriormente se desplegaron las imágenes desde una maquina EC2 en AWS en dos computadores distintos (PC1 y PC2).

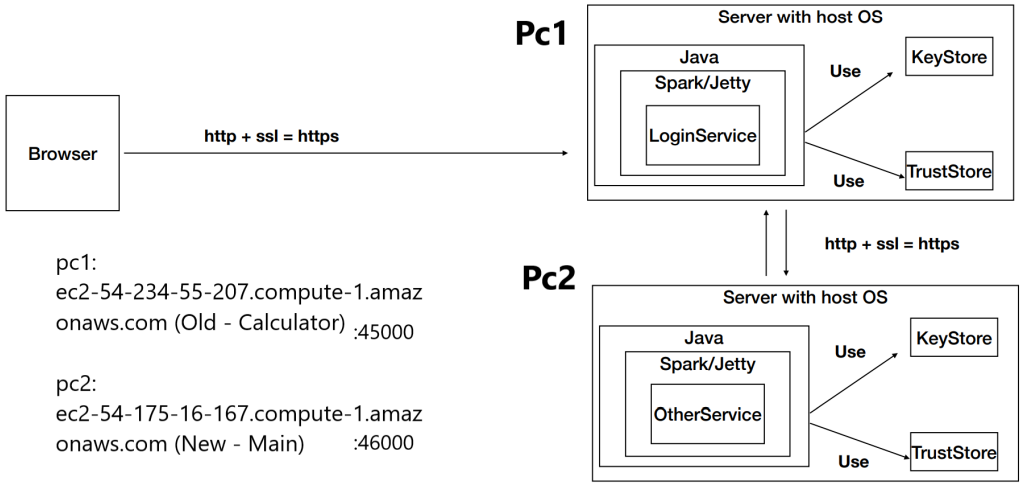

Figure 1: Modelo de componentes.

4 Conclusiones

- Se logra comprender el funcionamiento de AWS Y Docker
- Se logra rcomprender como realizar una aplicacion web segura con https, de forma que solo se puede acceder teniendo el certificado
- La seguridad es algo muy importante en nuestra aplicación web y es algo en lo que sebe profundizar

5 Pruebas

Como podemos ver, esto es lo que nos arroja el pc2 cuando hacemos la petición con los datos, mediante el http reader se va a retornar un string con lo que manda el Pc2, eso es lo que estamos imprimiendo y es lo que se va a mostrar en pantalla.

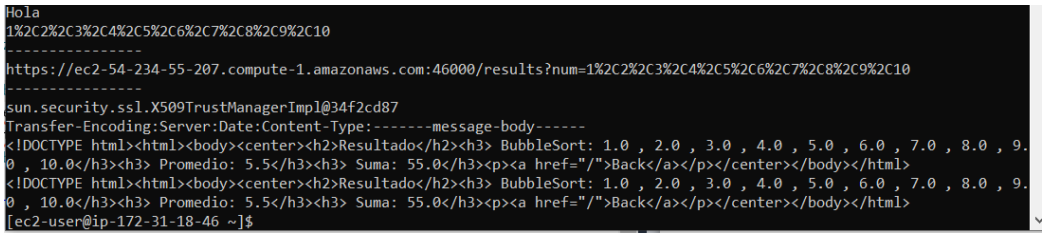

Figure 2: Prueba Pc1.

Dentro del Pc2 hay unos prints donde se ve que es lo que se solicita, así verificamos que efectivamente está haciendo la petición de Pc1 a Pc2.

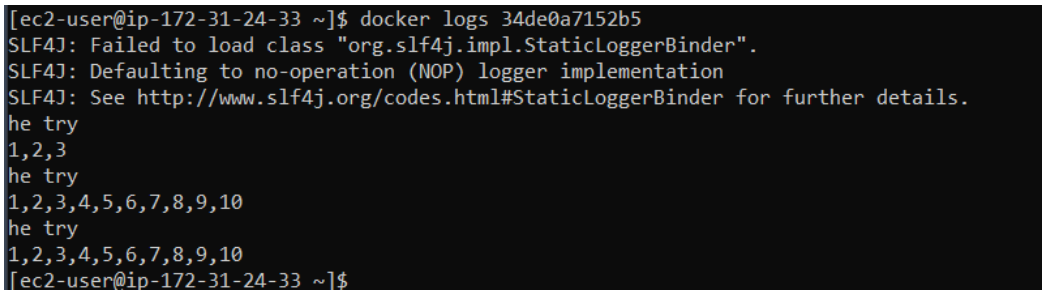

Figure 3: Prueba Pc2.

References

Repositorio https://github.com/luisalejandrojaramillo/AREPSecurityAppLab07