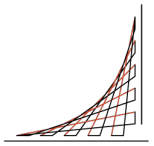


Laboratory 6

Safe Distributed Application on All Its Fronts

Author: Angi Paola Jimenez Pira
Teacher: Luis Daniel Benavides Navarro

Date: Friday 19th March, 2021
Bogota, Colombia

1 Introduction

The purpose of this workshop is to learn about security in web applications using tools as Java, keytool, Spark and Services of AWS, developing an architecture composed by differents servers and secured communications between those.

In this document you will find the explanation of challenge purposed in the laboratory 6, the architecture implemented, the tools used during development of this project and other sections that are important to understand how the project works (how install it, how run it, etc.) and if you want extend the code and implement new functionalities.

2 Objectives

- To develop an architecture using different tools for to ensure safety of the application.
- To learn about use of EC2 services of AWS.
- To learn about use of Keytool.
- To use known tools like Circleci, Maven, Java, Git, GitHub, Keytool and Spark to develop a secure web app.

3 Glossary

- **Keytool:** Tool used to manages a keystore (database) of cryptographic keys, X.509 certificate chains, and trusted certificates.
- **Spark Framework:** A micro framework for creating web applications in Kotlin and Java 8 with minimal effort.
- **AWS:** Amazon web service is an online platform that provides scalable and cost-effective cloud computing solutions.
- **EC2:** Amazon Elastic Compute Cloud (Amazon EC2) is a service that provides scalable computing capacity in the Amazon Web Services (AWS) Cloud.
- **Circleci:** CircleCI is a modern continuous integration and continuous delivery (CI/CD) platform.
- **Git:** Git is an open-source version control system that was started by Linus Torvalds—the same person who created Linux.
- **Github:** GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

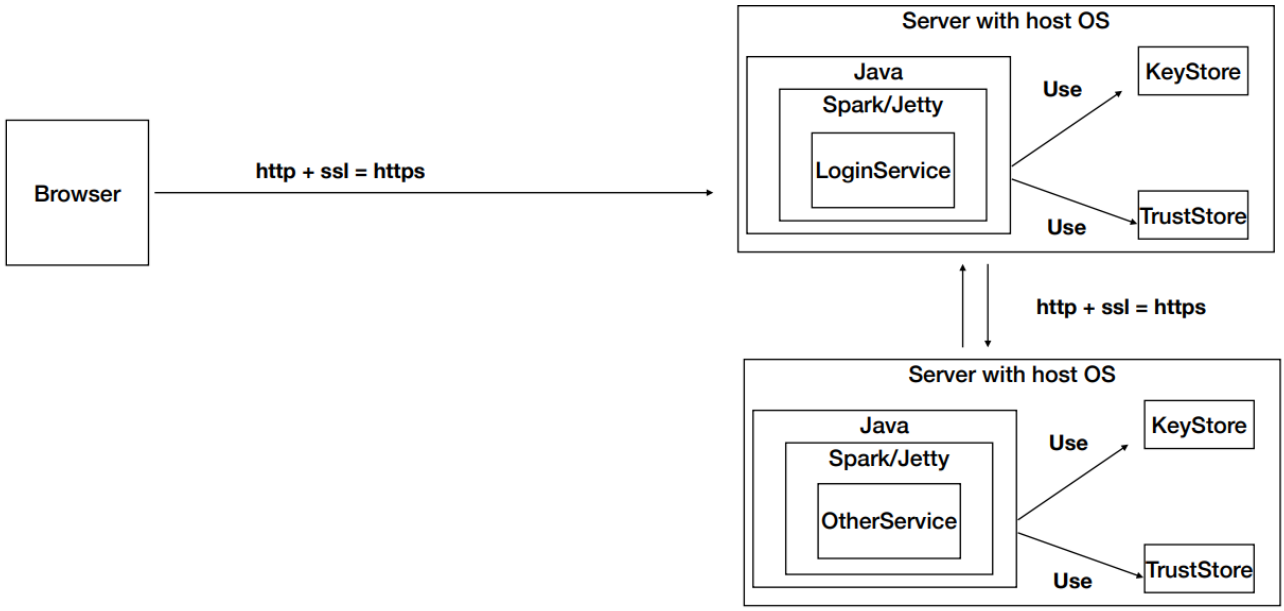
4 Challenge description

Develop a secure web application with the following requirements:

- It must allow secure access from the browser to the application. In other words, it must guarantee user authentication, authorization and integrity.
- It must have at least two computers communicating with each other and remote services access must guarantee: authentication, authorization and integrity between the services. No one can invoke the services if they are not authorized.
- Explain how you would scale your security architecture to incorporate new services.

5 Architecture of program

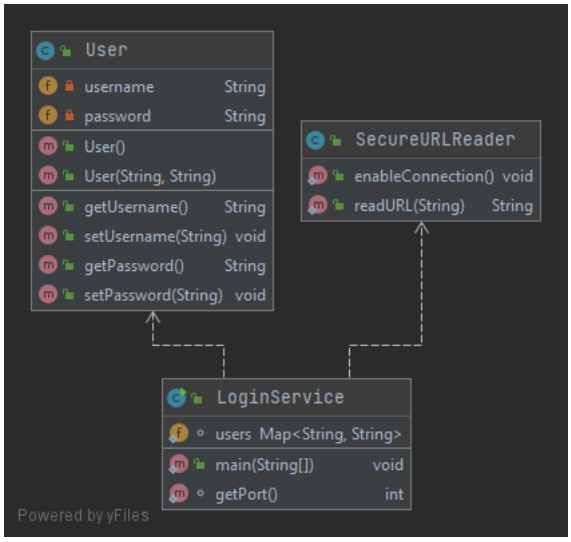
The next image shows the architecture implemented in this project:



5.1 Class Diagram

The next image shows the class Diagrams of the solutions implemented:

- **LoginService:**

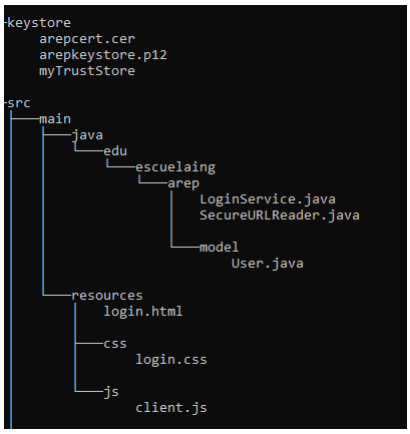


- **Other Service:**

5.2 Project Tree

The next image shows the structure of the projects generated by the Maven dependency manager, with the objective of giving more information for the reader about how the solutions were implemented:

- **LoginService:**



- **Other Service:**

5.3 Technology Stack

- **Development environment:** Java Development Kit 8, IntelliJ IDEA, Spark Framework, Keytool
- **Dependency Manager:** Maven
- **Version Control:** Git, Github
- **Tests:** JUnit4
- **Deploy and build:** EC2, Circleci

5.4 Documentation

The documentation and instructions for use are in a git repository: https://github.com/angipaola10/AREP-LAB6.

6 Conclusions

- The use of tools like Circleci to build a web app is very important because this tools help us to check the code and test developed in the program. Also, with this tools we can found bugs, duplicated code and other factors that make our project less efficient.
- Build web applications is a big responsibility because these manage important data and this information could can fall into the hands of malicious persons. For this reason is very important know different methods and tools to create secure applications web.

References

[1] Oracle. https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html

[2] SimpliLearn. https://www.simplilearn.com/tutorials/aws-tutorial/what-is-aws

[3] Circleci. https://circleci.com/docs/enterprise/overview/

[4] Spark. https://sparkjava.com/

[5] OpenSource. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html