ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

Escuela Colombiana de ingenieria Julio Garavito

Facultad de Ingeniería de Sistemas

APLICACIÓN DISTRIBUIDA SEGURA EN TODOS SUS FRENTES

ARQUITECTURAS EMPRESARIALES

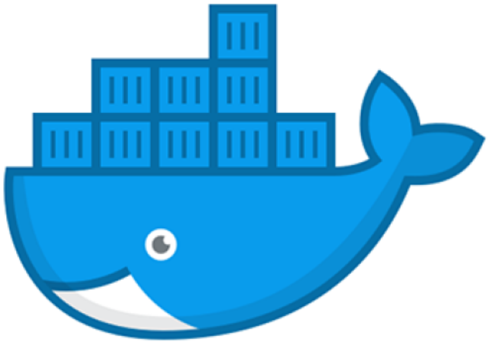
Autor:
Crhystian Camilo Molano Chacon

1 Introduccion

The workshop consists of creating a small web application using the Spark java microframework (http://sparkjava.com/). Once we have this application we will proceed to build a container for docker for the application and we will deploy and configure them on our local machine. Then we will close a repository on DockerHub and upload the image to the repository. Finally, we will create a virtual machine on AWS, install Docker, and deploy the container we just created.

2 Docker

Docker makes development efficient and predictable Docker eliminates mundane and repetitive configuration tasks and is used throughout the development lifecycle for fast, easy, and portable application development - desktop and cloud. Docker's comprehensive platform includes UI, CLI, API, and security that are designed to work together across the entire application delivery lifecycle.



3 AWS Educate

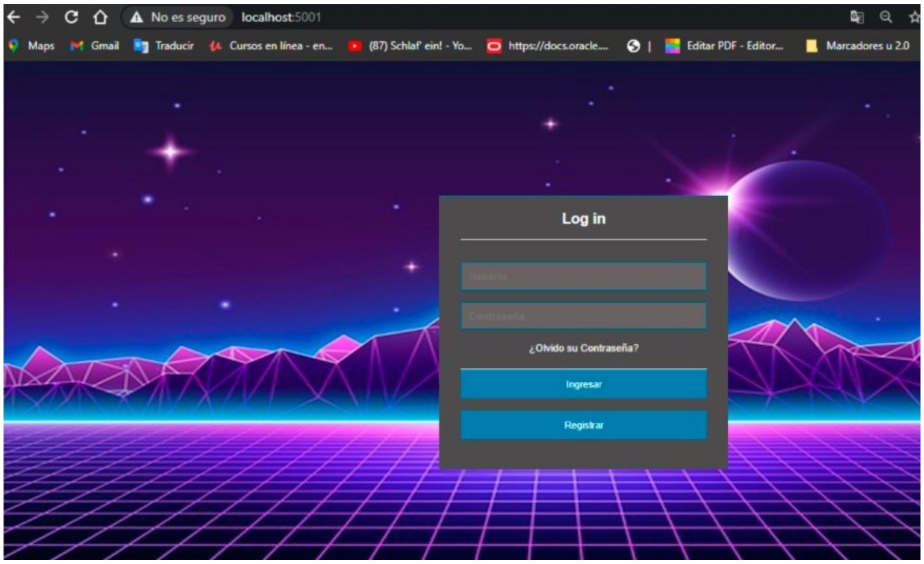
AWS Educate is an Amazon global initiative that aims to provide students with comprehensive resources to develop cloud skills. It is a no-cost program that provides access to AWS content, training, itineraries, and services.

4 Workshop

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 Tests

A login was created where it is handled safely and you can exercise your credentials and user safely there, identity classes were created with user details, as well as a url reader was created to read the url and its respective spark web.



execution join