

Tarea 1 Servicios en contenedores

José Carlos Núñez Valverde
jcnv21@gmail.com
Área académica de Ingeniería en Computadores
Instituto Tecnológico de Costa Rica
CE 4303 Principios de Sistemas Operativos

I. INTRODUCCIÓN

El presente documento brinda la descripción de un servidor alojado en contenedores, a través de Docker, con el fin de crear un ambiente controlado y aislar la ejecución del sistema operativo (SO). esto por medio de la creación de una imagen de Docker, la cual permitirá recibir imágenes (archivos .bmp) secuencialmente provenientes de un cliente local (host), utilizando sockets y aplicándoles un algoritmo para la detección del color predominante/principal de la imagen y así con su respectiva clasificación (roja, verde o azul) en una carpeta con el mismo nombre dentro del volumen host.

II. AMBIENTE DE DESARROLLO

A continuación se detallan las principales herramientas utilizadas para el desarrollo e implementación del proyecto.

II-A. Docker

Proyecto Open Source que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

II-B. Imagen Centos

CentOS distribución del sistema operativo Linux, el cual se utiliza por medio de una imagen de Docker en Docker Hub para la ejecución del servidor.

II-C. Visual Studio Code

Editor de código desarrollado por Microsoft para Windows, Linux y MacOS, utilizado para realizar/editar código, en este caso el lenguaje C.

III. DETALLES DE DISEÑO

III-A. Docker

Encargado de alojar al servidor para hacer uso de las funciones de aislamiento de recursos del kernel de Linux dando lugar a contenedores independientes, dentro del cual se ejecutará una única aplicación, este caso el servidor.

III-B. Cliente

Sistema encargado de interactuar con el usuario (personas) por medio de la terminal para enviar una imagen y la ip para ser ambas analizadas por el servidor y posteriormente a una nueva iteración volviendo a solicitar una nueva imagen para repetir el proceso, hasta que ingrese la palabra 'fin'.

III-C. Servidor

Programa / aplicación encargada de procesar la imagen y determinar el color principal de la misma, esto a través de la interacción con el contenedor en donde se encuentra.

III-D. Sockets

Protocolo de mensajería entre el Cliente y el Servidor, encargado de transferir la información necesaria para ejecutar todo el flujo de los programas (algoritmos).

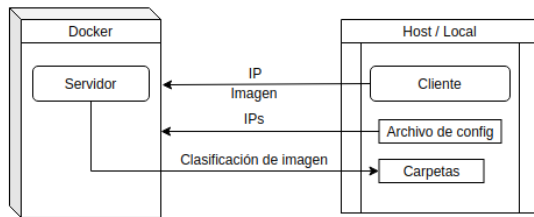


Figura 1: Diagrama de flujo de datos

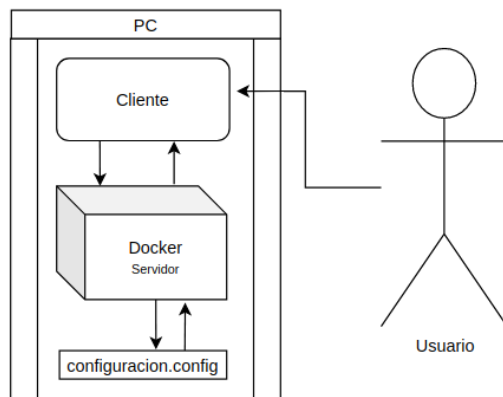


Figura 2: Diagrama de Arquitectura

IV. INSTRUCCIONES DE CÓMO SE UTILIZA EL PROYECTO

A continuación se detalla los pasos para utilizar el programa:

Localmente

1. Abrir terminal
2. Posicionarse en el folder del proyecto
3. Ejecutar: **make**
- Servidor -
4. Ejecutar: **docker build -t is .**
5. Ejecutar: **docker run --publish 8000:8080 -v <path_local>:/server-storage/ --detach --name cs is**
- Cliente -
6. Ejecutar: **./client <ip>**
7. Ingresar ruta de imagen
8. Repetir paso anterior
9. Finalizar, ingresar fin como ruta de imagen.

Desde Docker Hub

1. Abrir terminal.
2. Posicionarse en el folder del proyecto.
- Servidor -
3. Ejecutar: **docker pull josek21/tarea1**
4. Ejecutar: **docker run --publish 8000:8080 -v <path_local>:/server-storage/ --detach --name cs josek21/tarea1** - Cliente -
5. Ejecutar: **./client <ip>**
6. Ingresar ruta de imagen
7. Repetir paso anterior
8. Finalizar, ingresar fin como ruta de imagen.

V. CONCLUSIONES

La utilización de Docker para crear contenedores y así aislando responsabilidades a otros ambientes es de mucha ayuda, ya que aunque existen algunas desventajas como por ejemplo, asuntos de seguridad, el mismo trae muchas características positivas de las cuales podemos hacer uso y así facilitarnos nuestras vidas, algo que al final todos buscamos.

VI. SUGERENCIAS Y RECOMENDACIONES

- Tener el SO (imagen) descargado, esto para que cada vez que se crea un contenedor a partir de ese SO no deba descargarlo de nuevo, sino que use el que tiene en cache.

REFERENCIAS

- [1] P. McKee. How to Build and Test Your Docker Images in the Cloud with Docker Hub, Tomado de : <https://www.docker.com/blog/how-to-build-and-test-your-docker-images-in-the-cloud-with-docker-hub/>
- [2] Docker Hub. Quickstart, Tomado de : <https://docs.docker.com/docker-hub/>
- [3] Geekforgeeks. Socket Programming in C/C++, Tomado de : <https://www.geekforgeeks.org/socket-programming-cc/>
- [4] Prathiyus. Sending image through sockets, Tomado de : <https://www.daniweb.com/programming/software-development/threads/454598/sending-image-through-sockets>