

Despliegue de aplicaciones contenerizadas: un caso de implementación basado en Crane

José Miguel Silva Pavon¹, Franco Bellino¹, Patricia Bazán³, Alejandra B. Lliteras^{2,4}, Jose Arcidiacono¹, Nicolás del Río¹
UNLP, Facultad de Informática¹, LINTI² UNLP, Facultad de Informática, LIFIA³ UNLP, Facultad de Informática, LINTI,⁴ CICPBA
js.silva.010@gmail.com, fran85bellino@gmail.com, pbaz@info.unlp.edu.ar, alejandra.lliteras@lifia.info.unlp.edu.ar,
jarcidiacono@linti.unlp.edu.ar, ndelrio@info.unlp.edu.ar

RESUMEN

Crane es una herramienta para el despliegue local de aplicaciones en contenedores, liviana, de propósito general y con capacidades de escalado automático, lo que la diferencia, por ejemplo, de la herramienta Minikube, la que permite algunas pruebas locales de la API de Kubernetes y se utiliza principalmente para el desarrollo de nuevas funcionalidades para K8s.

El diseño de Crane presentado en [Arcidiacono et al., 2022] contempla la implementación de la herramienta con un componente de back-end para la creación y despliegue de contenedores y un componente de front-end orientado a un usuario final que debe desempeñar el rol de devop dentro de un proyecto [Httermann M, 2012].

El objetivo de este trabajo es proporcionar un servicio REST [Fielding & Taylor, 2002] para crear y desplegar servicios docker con sus correspondientes herramientas de medición, definición de políticas de escalado y gestión de alertas para la toma de decisiones.

CONTEXTO

Este proyecto surge con el fin de implementar una herramienta para el despliegue local de aplicaciones en contenedores, liviana, de propósito general y con capacidades de escalado automático.

DEHIA es un administrador de flujo de trabajo para la recopilación de datos con intervención humana [Arcidiacono, 2020]. Su arquitectura se basa en microservicios. En un primer intento de uso de DEHIA para la entrega en un servidor local, resultó ser complejo y difícil de replicar debido a sus diversos componentes y tecnologías.

Una posible solución a este problema es la contenerización [Bullington-McGuire et. al, 2020]. Para comenzar, se eligió el componente más simple, una puerta de enlace. Este componente no tiene dependencias funcionales con los otros componentes y no tiene estado interno.

El componente de puerta de enlace sólo necesita un puerto abierto (para recibir las solicitudes que tiene que redirigir) y espera un pequeño conjunto de parámetros. Por eso, era viable implementarlo automáticamente. Para ello se decidió desarrollar una herramienta de despliegue automático de Docker (denominada "Crane"), con el añadido de escalar la aplicación bajo demanda, creando nuevas instancias.

FORMACIÓN DE RECURSOS HUMANOS

- Desarrollar conocimientos en Docker y el manejo de contenedores, así como, conceptos de seguridad, comunicación entre contenedores o simplemente ser capaz de analizar las métricas obtenidas por un determinado contenedor y tomar decisiones fundamentadas.

- Conocimientos en la gestión y optimización de la infraestructura de la aplicación, tales como: balanceos de carga para optimizar al máximo los recursos disponibles, gestión de DNS, escalabilidad y también monitoreo

- Aprendizaje en la recolección de métricas de rendimiento de los sistemas, manejo de alertas personalizadas que en caso de ser activadas, responderán dependiendo de unas políticas definidas.

- Conocimientos en la creación de contenedores, su comportamiento en cada momento y qué se puede hacer para mejorar su rendimiento o qué hacer en caso de fallo.

- Uso de CRANE para acelerar el desarrollo de aplicaciones, lo que permitirá poder tener desplegado rápidamente un entorno, mediante una simple petición http.

LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

- Creación de aplicaciones seguras, con reglas claramente definidas que hagan transparente al desarrollador la complejidad del despliegue que suele tener una aplicación distribuida.

- Implementación de un API REST que interactúa con Crane enviando en el cuerpo de la solicitud los nombres de las imágenes del repositorio oficial de docker que desea utilizar y crea, almacena y despliega un stack de servicios a partir de un archivo docker compose.

- Definición de reglas y políticas para la toma de decisiones pudiendo así ajustar el consumo que realiza el servicio, sin desperdiciar/sobrecargar la infraestructura cuando el rendimiento baja/sube.

- Integración con un sistema de gestión de alertas y con un sistema de gestión de políticas, creación de archivos de configuración dinámicos y estáticos para despliegue de contenedores, la administración de las configuraciones de seguridad de los contenedores, la implementación de base de datos para la gestión de aplicaciones y el modelo RBAC para la autenticación y autorización sobre el sistema (Figura 1)

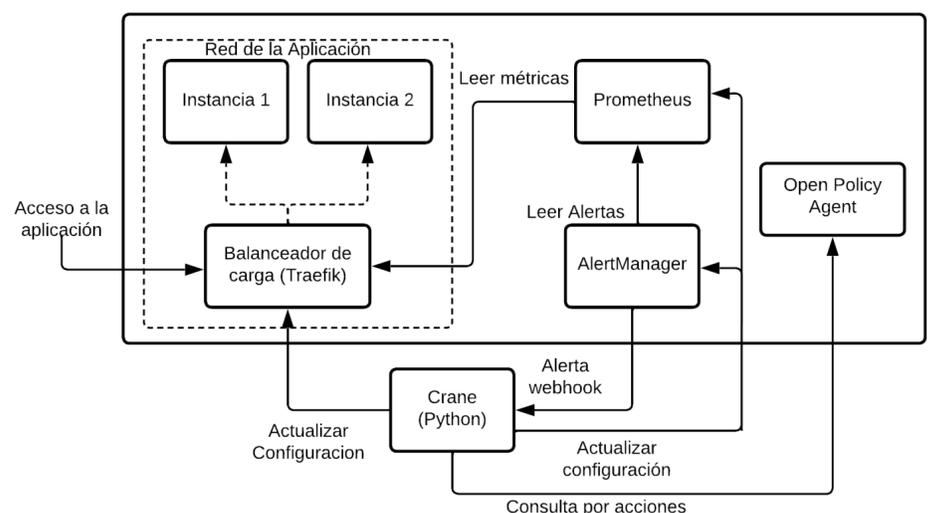


Figura 1.- Arquitectura instanciada en Crane

REFERENCIAS

[Arcidiacono et al., 2022] Arcidiacono, J., Bazán, P., del Río, N., & Lliteras, A. B. (2022). Crane: A Local Deployment Tool for Containerized Applications. In Conference on Cloud Computing, Big Data & Emerging Topics (pp. 58-71). Springer, Cham.

[Httermann M, 2012] Httermann, M. (2012). DevOps for developers. Apress: delivers a practical, thorough introduction to approaches, processes and tools to foster collaboration between software development and operations.

[Fielding & Taylor, 2002] Fielding, R. T., & Taylor, R. N. (2002). Principled design of the modern web architecture. ACM Transactions on Internet Technology (TOIT), 2(2), 115-150.

[Arcidiacono, 2020] Arcidiacono, J. (2020). DEHIA: una plataforma liviana para definir y ejecutar actividades con intervención humana basadas en workflows (Doctoral dissertation, Universidad Nacional de La Plata).

[Bullington-McGuire et. al, 2020] Bullington-McGuire, R. and Dennis, A.K. and Schwartz, M. (2020). Docker for Developers: Develop and run your application with Docker containers using DevOps tools for continuous delivery. Packt Publishing.