

Implementación y evaluación de un sistema de orquestación de contenedores para alta disponibilidad en procesamiento distribuido

Descripción del proyecto

Las tecnologías de contenedores se usan hoy en día ampliamente en el desarrollo de software gracias a que estas brindan entornos de computación modulares, autocontenidos y reproducibles. El objetivo de este proyecto es validar la característica de **disponibilidad** que pueden ofrecer las tecnologías de orquestación de contenedores (e.g. Kubernetes y Docker Swarm) a la hora de desarrollar y desplegar aplicaciones basadas en contenedores.

Aplicación propuesta

Su grupo debe desarrollar una aplicación que permita la gestión de tareas y proyectos. Un proyecto puede ser “Elaboración de un sistema de información para el puesto de salud XXX” y una tarea podría ser, levantamiento de requerimientos. En general, puede ser cualquier proyecto y definir cualquier cantidad de tareas para cada proyecto. Tanto los proyectos como las tareas tienen fechas de creación y fechas de terminación. El aplicativo se debe encargar entonces de hacer las operaciones CRUD sobre proyectos y tareas. Todas las operaciones de gestión que se hagan sobre los proyectos y las tareas (creación, modificación, borrado y recuperación) deberán quedar consignadas en un archivo de log. Importante si un proyecto se borra, también todas sus tareas son borradas de manera automática.

Se espera

- Que los proyectos y las tareas se deben hacer persistentes en una base de datos relacional.
- Que el diseño de la aplicación debe hacerse altamente modular de modo que cada uno de los módulos sean desplegados en un contenedor.

No se requiere

- Que la aplicación gestione usuarios
- Que tenga un mecanismo de autenticación
- De una interfaz de usuario tampoco muy sofisticada.

Objetivos

- Diseñar e implementar una aplicación web que permita la gestión de proyectos y de tareas asociadas a un proyecto. Los diferentes módulos de la aplicación deben ser accesibles tanto por la web como a través de APIs.
- Desplegar el prototipo de su aplicación a través de un orquestador de contenedores que bien puede ser Kubernetes o Docker Swarm.
- Generar y ejecutar una batería de pruebas usando la herramienta Locust¹ o JMeter donde se pongan a prueba los diferentes endpoints de los diferentes módulos de su aplicación. (Generar escenarios de fallos para evaluar la disponibilidad del sistema)
- Análisis de los datos observados.

Rúbrica de evaluación

Criterio	Excelente	Bueno	Aceptable	Insuficiente
[20] Documentación y Calidad de la Presentación en video	[20] El informe es claro y completo y sin errores gramaticales. Presentación bien estructurada, se evidencia la participación de todos los miembros del grupo y el video tiene una duración no superior a 20 minutos.	[15] El informe es mayormente claro con algunos errores menores. Presentación adecuada, todos los miembros del grupo participan en la presentación y el video tiene una duración no superior a 20 minutos.	[10] El informe cumple con los requisitos mínimos, se evidencian errores. La presentación presenta algunas falencias, o no hay evidencia de participación de todos los miembros del grupo o superan los 20 minutos.	[5] El informe es desorganizado o incompleto; la presentación es pobre.

¹ <https://locust.io/> (visitado: 2024/05/09)

[30] Diseño e Implementación del Sistema	[30] Diseño innovador y alineado con objetivos; implementación robusta (validaciones de errores) y eficiente (evidencia de algoritmos eficientes).	[22] Diseño adecuado y cumplimiento de la mayoría de los objetivos; implementación correcta con algunos errores.	[15] Diseño e implementación básicos, cumplen objetivos mínimos con aspectos por mejorar.	[8] Diseño no cumple con los objetivos; implementación inadecuada.
[25] Pruebas y Análisis de Resultados	[25] Pruebas exhaustivas y análisis detallado, ofreciendo detalles claros.	[18] Pruebas adecuadas y buen análisis que ofrecen una evaluación razonable.	[12] Pruebas básicas, análisis superficial.	[6] Pruebas y análisis insuficientes o inadecuados.
[15] Escalabilidad y Disponibilidad	[15] El sistema muestra excepcional escalabilidad y disponibilidad bajo diferentes condiciones.	[11] El sistema muestra buena escalabilidad y disponibilidad con algunos problemas menores.	[7] El sistema muestra escalabilidad y disponibilidad limitadas.	[3] El Sistema no cumple con los requisitos de escalabilidad o disponibilidad.
[10] Innovación y Creatividad	[10] Proyecto muy innovador y creativo.	[7] Signos de creatividad, aunque sigue soluciones existentes.	[4] Proyecto cumple con requisitos, poca innovación.	[1] Falta de creatividad e innovación.